THIRD ANNUAL

2003 Microsoft® **STRATEGIC ARCHITECT FORUM**

for Customers

Envisioning the Service-Oriented Enterprise: Metropolis—Part 1

Pat Helland
Architect
.NET Enterprise Architecture Team
Microsoft Corporation

Dave Campbell
Product Unit Manager
SQL
Microsoft Corporation

Outline

- Metropolis: Part 1
  - Metropolis: The Analogy
  - Practical Advice for Building Services
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

Outline

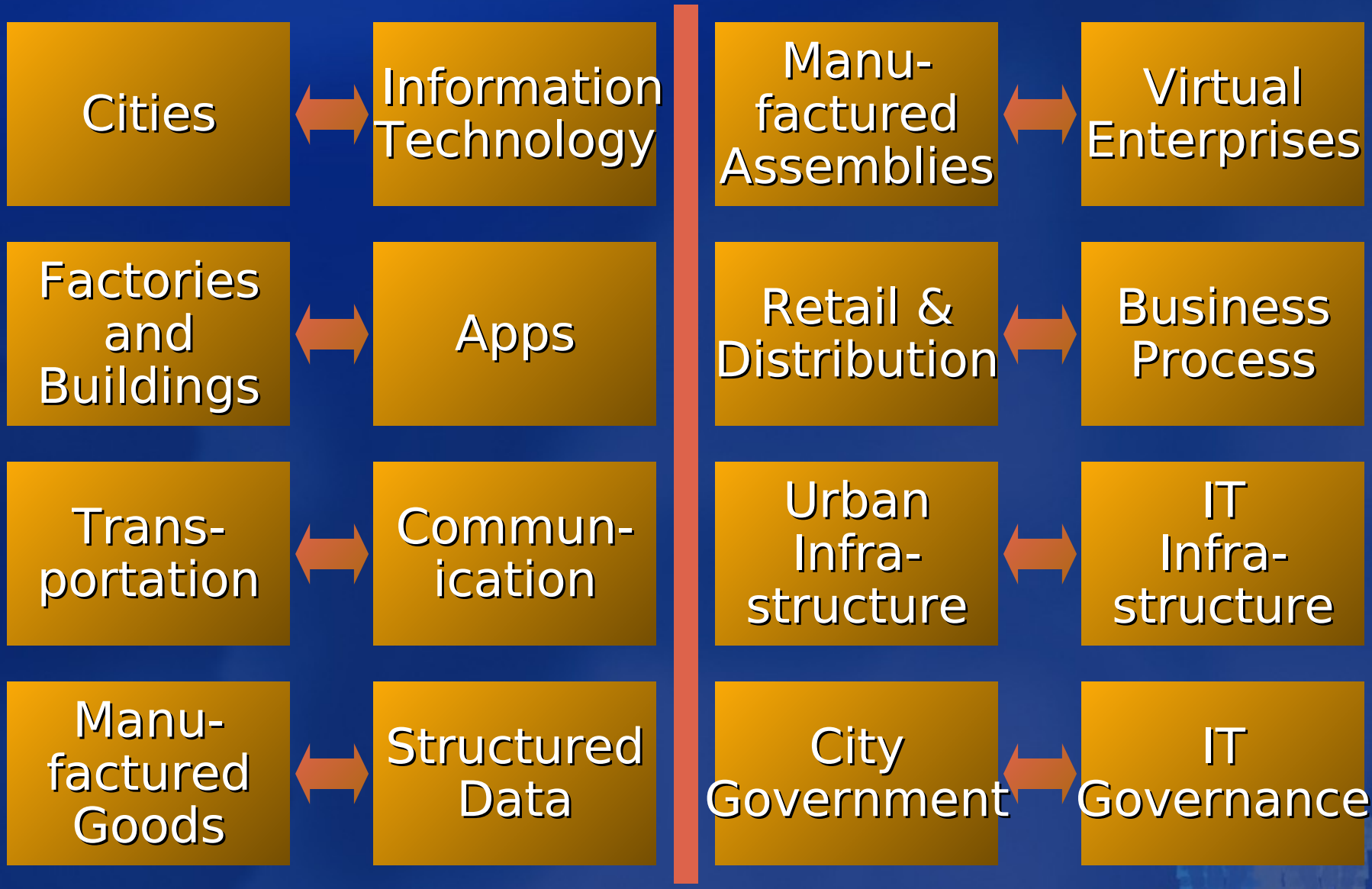
- Metropolis: Part 1
  - **Metropolis: The Analogy**
  - Practical Advice for Building Services
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

The Metropolis Analogy

- IT Shops and Cities
  - Gradual Evolution
  - Developed in Isolation
- Railroads Arrive
  - People Travel
  - Stuff Travels
    - Commodity versus Manufactured
- Internet Arrives
  - People Browse
  - Data Moves
    - Commodity versus Structured

We propose that this analogy shows us a lot about where we are heading!

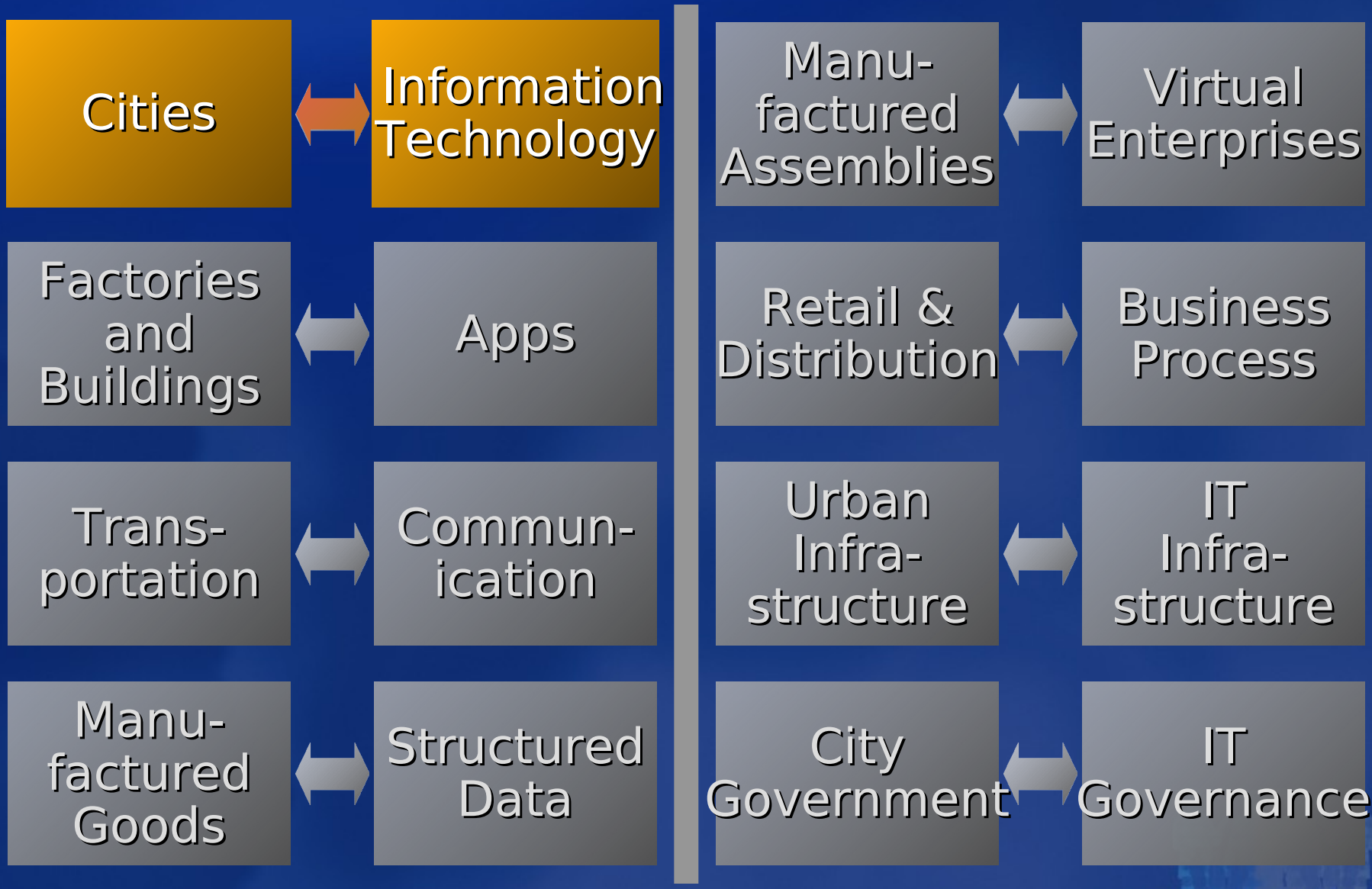
Metropolis


Cities
Information Technology
Manu-factured Assemblies
Virtual Enterprises
Factories and Buildings
Apps
Retail & Distribution
Business Process
Trans-portation
Commun-ication
Urban Infra-structure
IT Infra-structure
Manu-factured Goods
Structured Data
City Government
IT Governance

Implications of Metropolis

- Heterogeneity Happens!
- Ongoing IT Investment
  - Infrastructure versus Business
  - Historic Monuments
- Standardization Is Nascent
  - Connection Largely by People
  - Efficiencies Still to Come
- Business Process Is Nascent
  - Still Mostly Ad-hoc
  - Growing to Become Dominant Force
- Loose Coupling Helps Investments

Metropolis

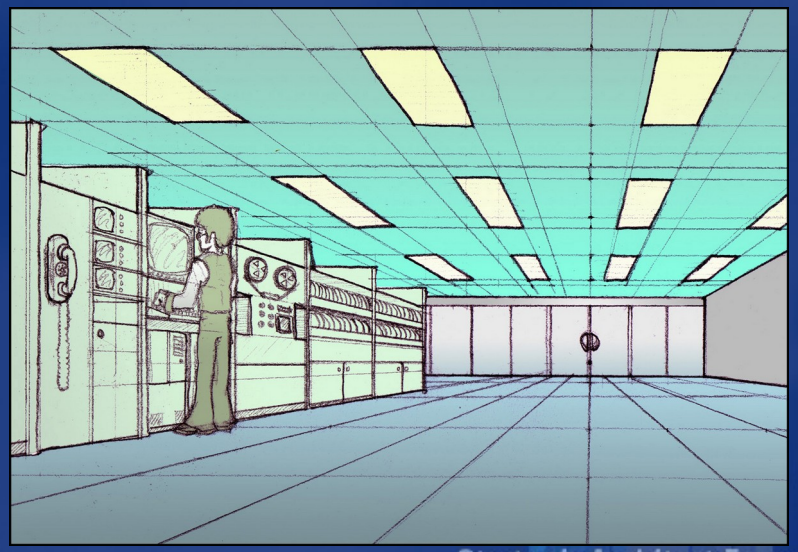

Cities
Information Technology
Manu-factured Assemblies
Virtual Enterprises
Factories and Buildings
Apps
Retail & Distribution
Business Process
Trans-portation
Commun-ication
Urban Infra-structure
IT Infra-structure
Manu-factured Goods
Structured Data
City Government
IT Governance

The Evolution of Cities

- Gradual Growth
  - Gather for Commerce and Manufacturing
- Independent Buildings
  - No Connections
- Independent Cities
  - Travel Too Hard
  - Cities Did Things Their Own Way

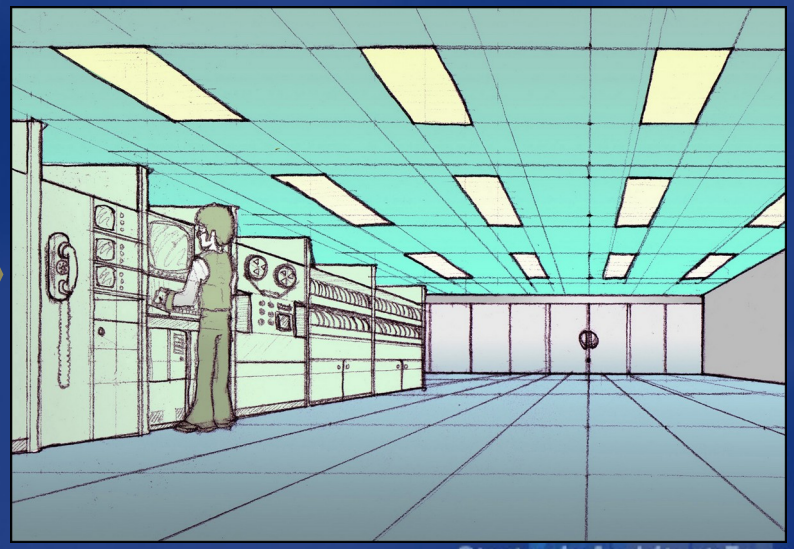
The Evolution of the IT Shop

- Gradual Growth
  - New Apps Gradually Built
- Applications Largely Independent
  - People Interact Separately with Apps
- B2B Still Limited
  - Largely via People
  - Independently Designed and Incompatible Apps

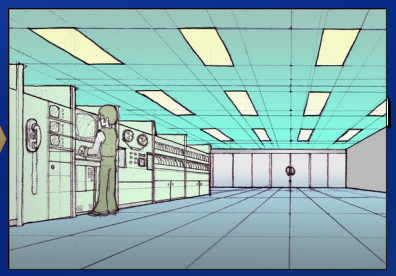


Cities and IT Shops

- Big Complex and Evolving Environments
  - Fed by Economics
  - Ongoing Investment
    - New and Renovated Buildings (or Apps)
    - Infrastructure for Connectivity
  - Both Have Historic Monuments to Consider!



Metropolis



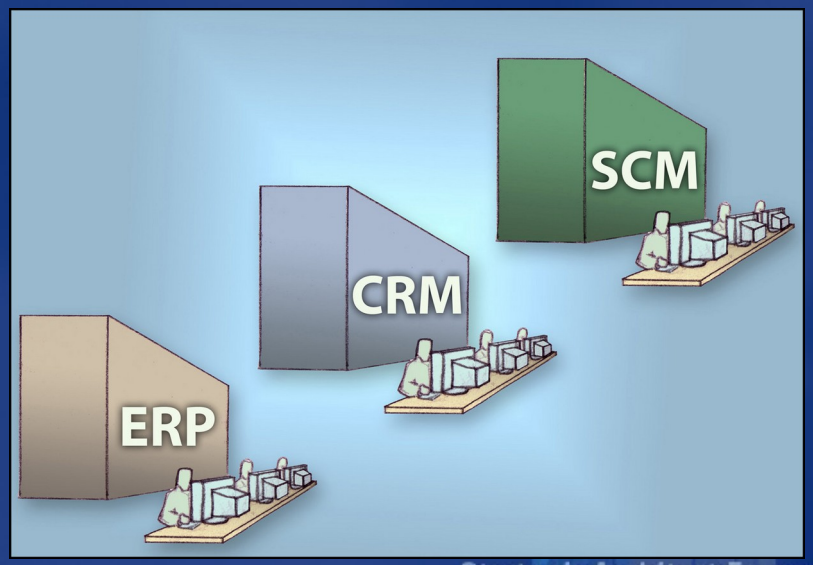
Factories and Buildings	Apps	Manu-factured Assemblies	Virtual Enterprises
Retail & Distribution	Business Process		
Trans-portation	Commun-ication	Urban Infra-structure	IT Infra-structure
Manu-factured Goods	Structured Data	City Government	IT Governance

Factories and Buildings

- Early 1800s
  - Simple Manufacturing
  - Independent
  - Shipping Difficult
- Late 1800s
  - Sophisticated Manufacturing
  - Autonomous Yet Interconnected
  - Located by Railroads
  - Attached to City Infrastructure

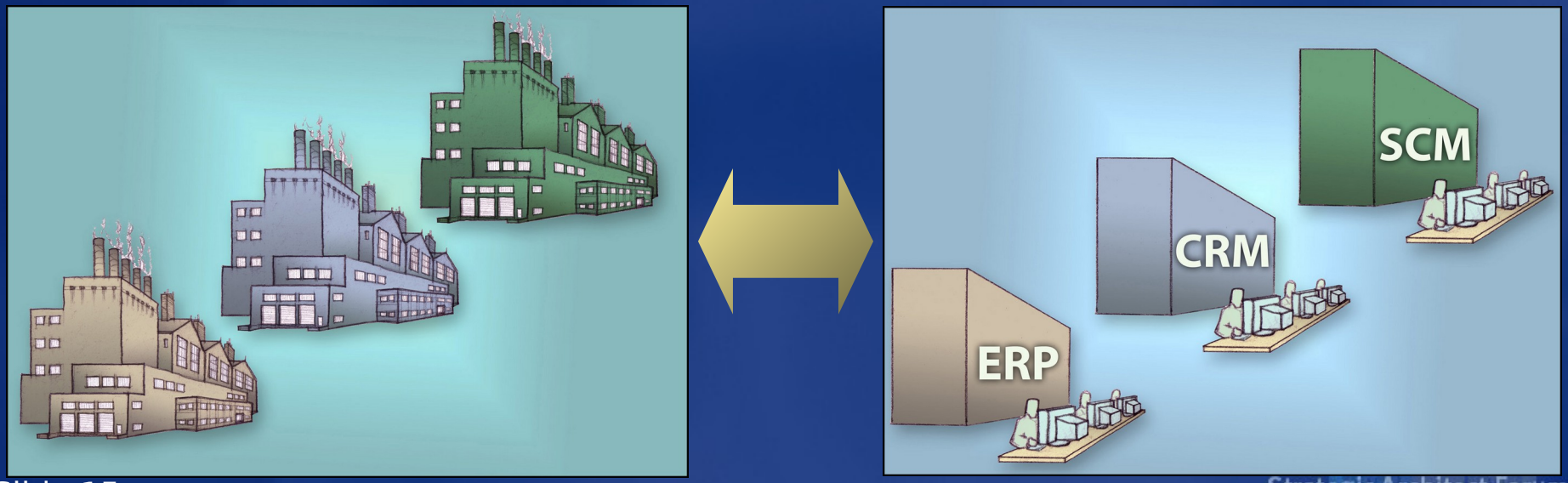
Applications

- Today
  - Produce Processed Data
  - Are Mostly Independent
    - Useful Interconnection Difficult
  - IT Infrastructure Nascent
  - Automated Business Process Nascent
- Tomorrow
  - Rich Interconnectivity
    - Sophisticated Data Interchange
    - Sophisticated Business Process
  - Tapped into IT Infrastructure

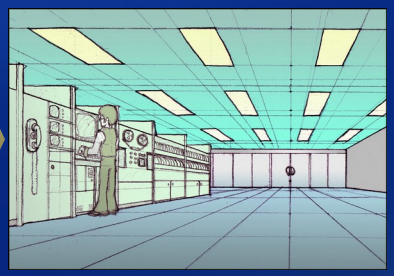


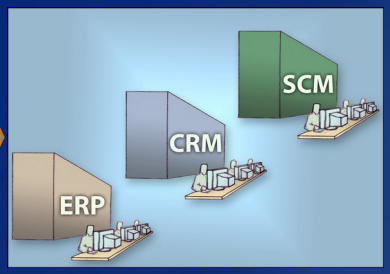
Factories and Apps

- Independence Is Essential
  - Get Work Done
  - Decoupled Evolution
- Advantages to Interconnection
  - Leverage Others
- Tap into Infrastructure
  - Services from City or IT Shop



Metropolis





Trans-portation ↔ Commun-ication

Manu-factured Goods ↔ Structured Data

Manu-factured Assemblies ↔ Virtual Enterprises

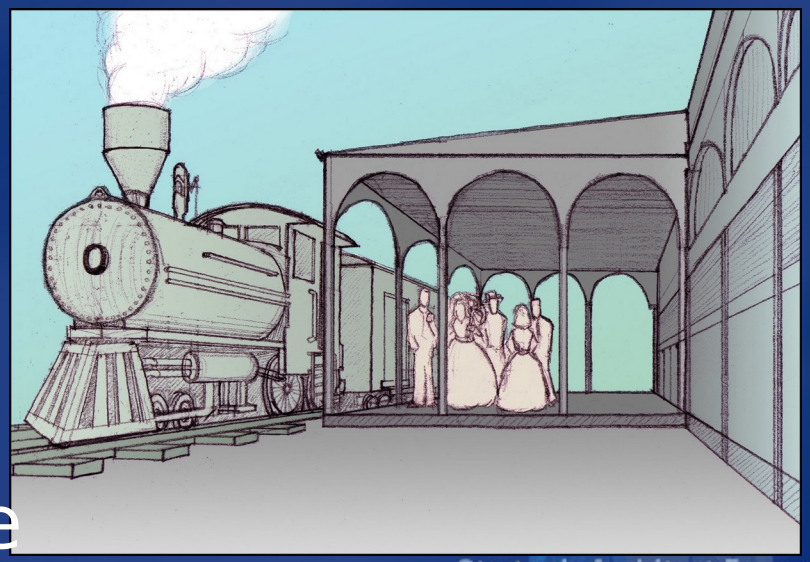
Retail & Distribution ↔ Business Process

Urban Infra-structure ↔ IT Infra-structure

City Government ↔ IT Governance

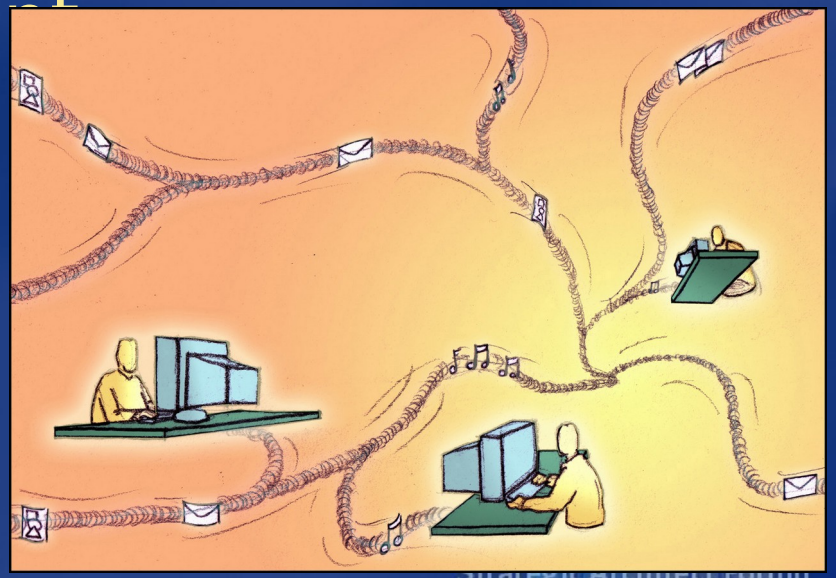
Transportation

- Railroad Arrives!
  - Money Moving People and Raw Commodities
  - Speculations, Booms, and Busts…
- Movement of People Changed Things
  - Retail Expanded
- Movement of Stuff Changed Things
  - Stuff Had to Work Together
  - Retailers Could Gather Stuff for Sale

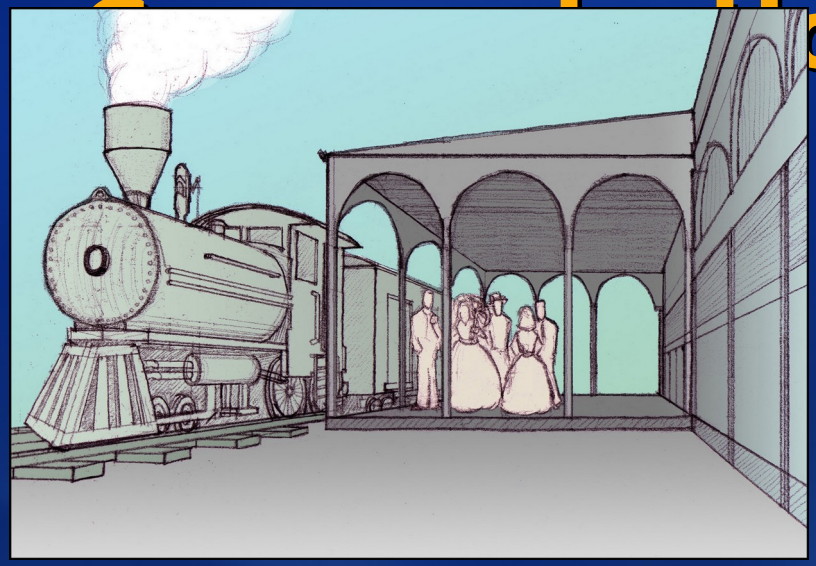


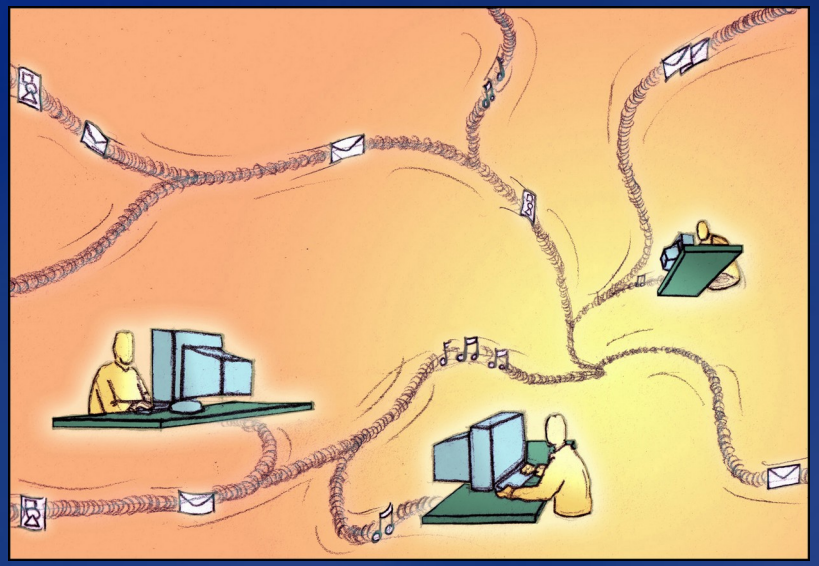
Communication

- Internet Arrives!
  - Money in Browsing and Moving Raw Data
  - Speculations, Booms, and Busts…
- People Browsing Changed Things
  - Directly Access Remote Apps
  - Driving Demand for Business Process
- Changes from Movement of Data Nascent
  - Data Still Doesn't Work Together
  - Business Process Still Very Limited



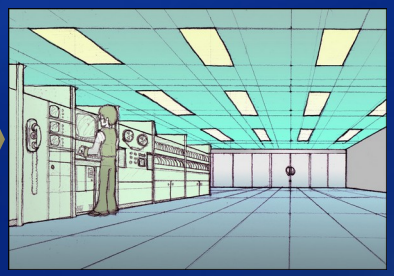
Transportation and Communication





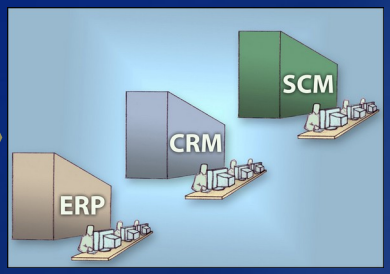
- Started Moving People and Commodities
- Drove New Changes
  - Standardization
    - Stuff and Data
  - Retail and Business Process
  - Economic Consolidation
    - Cities
    - IT Shops Using Structured Data

Metropolis



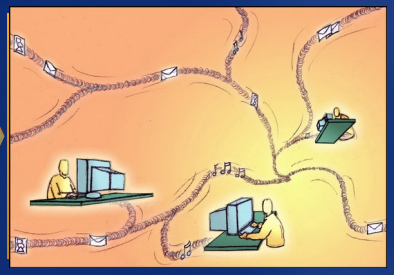
Manu-factured Assemblies

Virtual Enterprises



Retail & Distribution

Business Process



Urban Infra-structure

IT Infra-structure

Manu-factured Goods

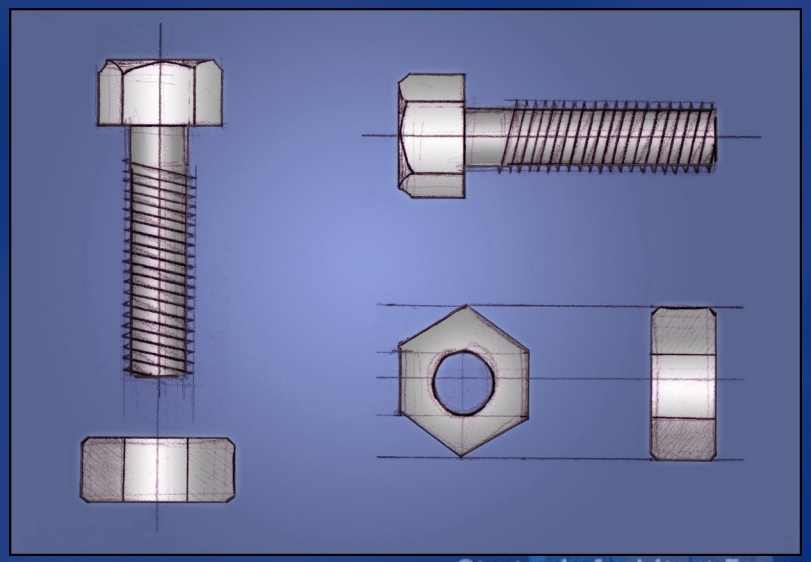
Structured Data

City Government

IT Governance

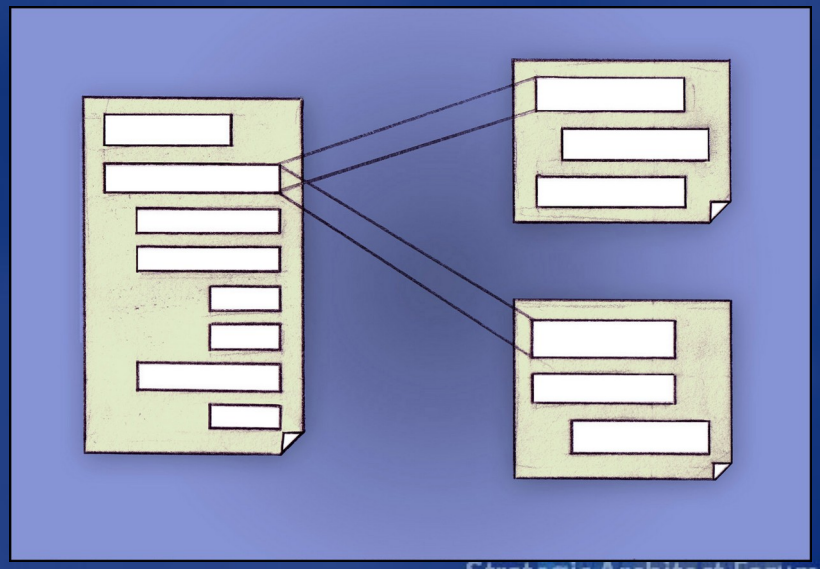
Manufactured Goods

- Early 1800s
  - Hand-crafted Goods
  - “Trim-and-Shim”
  - Eli Whitney—Interchangeable Parts
- Late 1800s
  - De Facto Standards per Industry
  - Marketplace Demanded Compatibility
  - Retailing Demanded Compatibility
- Retooled or Went Under!

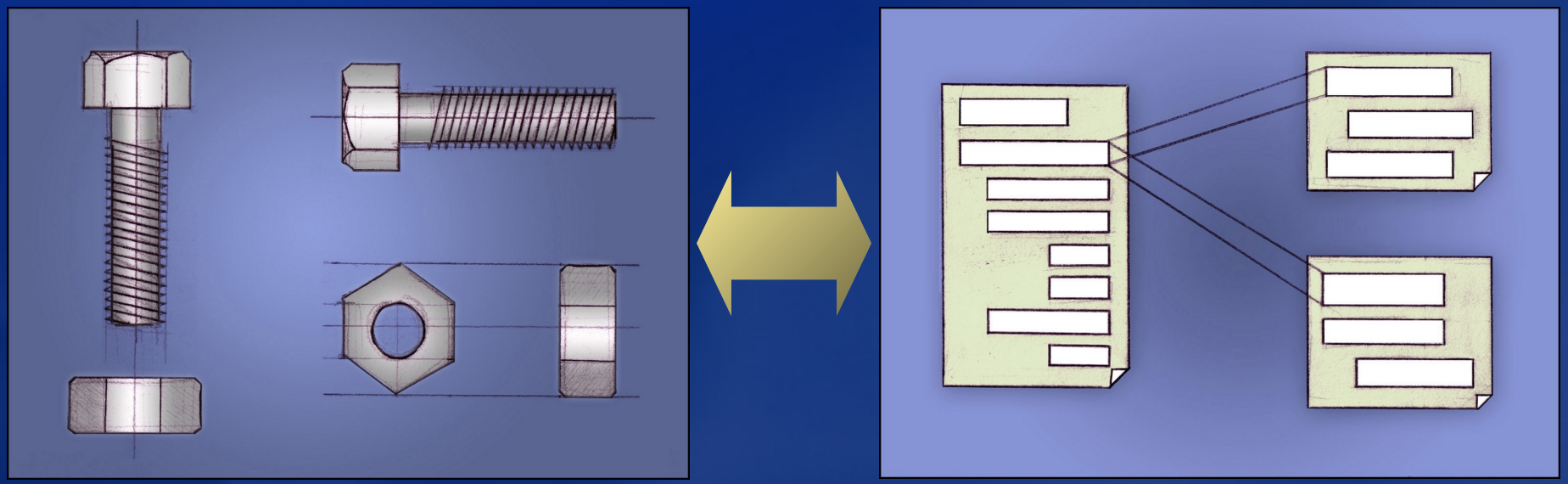


Structured Data

- Today
  - Non-Standard Data Structures
  - Mostly Human “Trim-and-Shim” to Integrate
  - Beginnings of Standardization
    - Web Services Foundation
    - Need Industry Standards
    - Demand for Business Process
- Soon
  - Industry De Facto Standards
  - Increased Compatibility
  - More Sophisticated Business Process

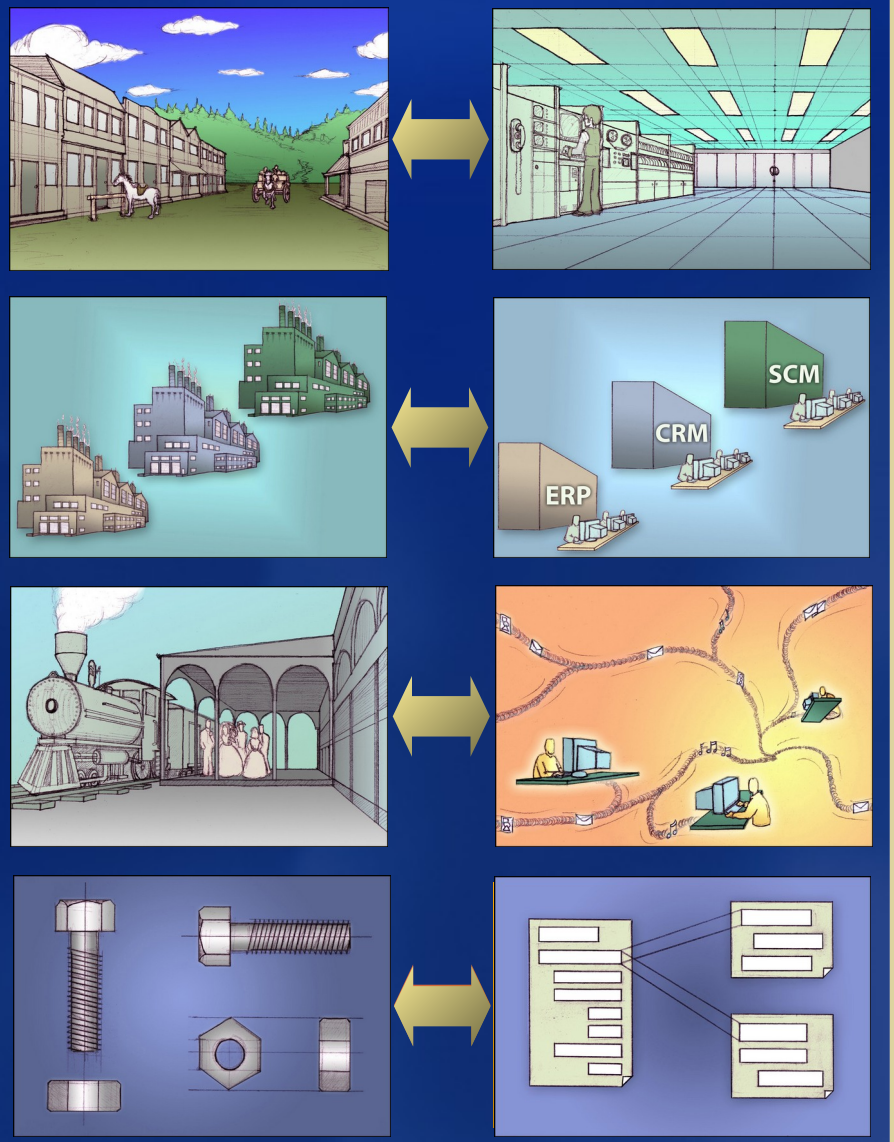


Manufactured Goods and Structured



- Must Connect to Other Stuff
  - Can't Live in Isolation
  - Manufacturing Retooled
    - New Efficiencies and Markets Came
  - Applications Must Retool
    - Data and Business Process Integration
    - Tremendous Payoffs to Come

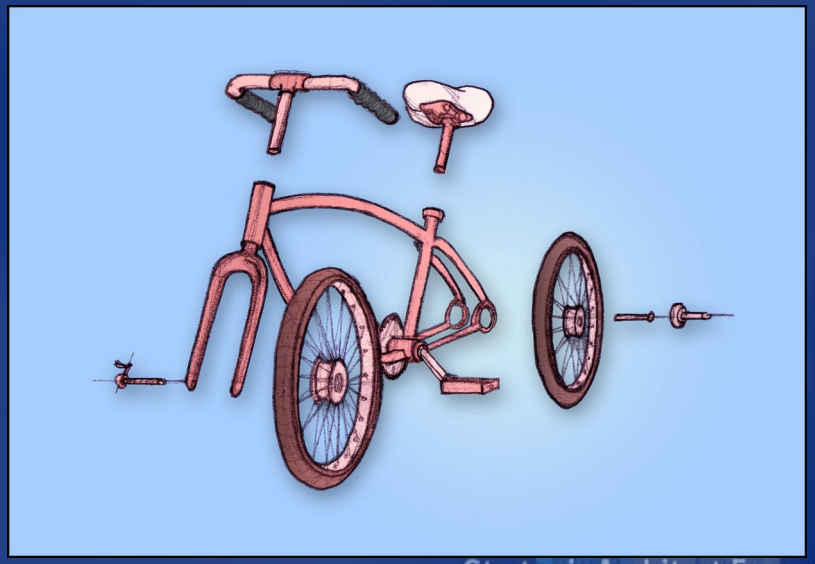
Metropolis



Manu-factured Assemblies	Virtual Enterprises
Retail & Distribution	Business Process
Urban Infra-structure	IT Infra-structure
City Government	IT Governance

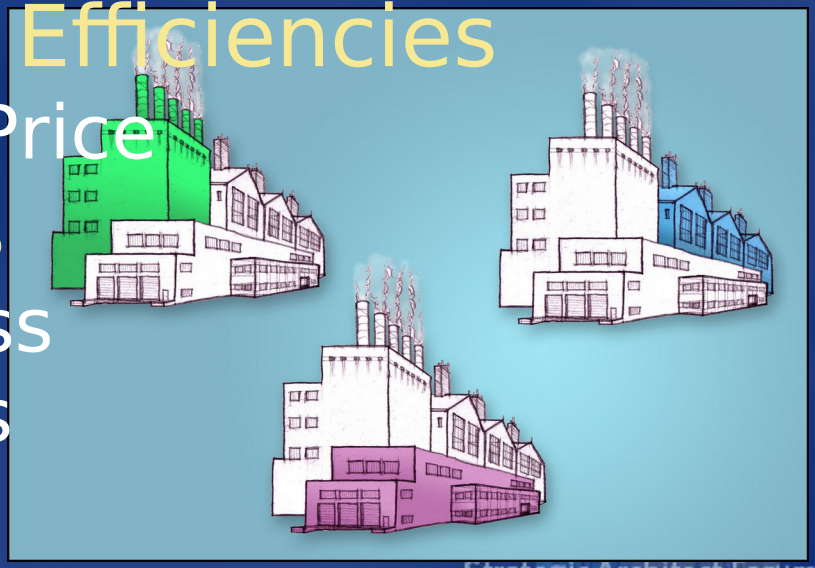
Manufactured Assemblies

- Assemble Parts from Other Factories
  - Requires Detailed Standardization
  - Produces High-Value Goods
  - Leverage Other Companies' Stuff
- Competition Drove Efficiencies
  - Better Parts
  - Better Prices



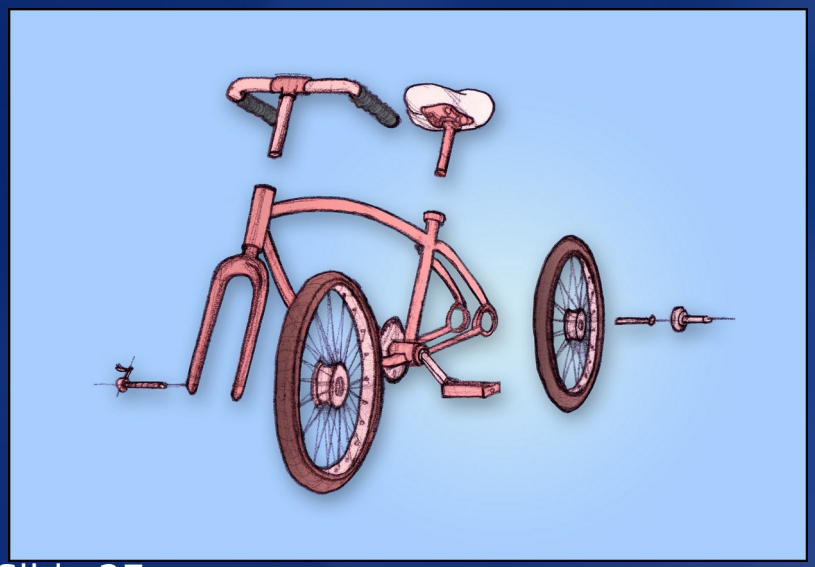
Virtual Enterprises

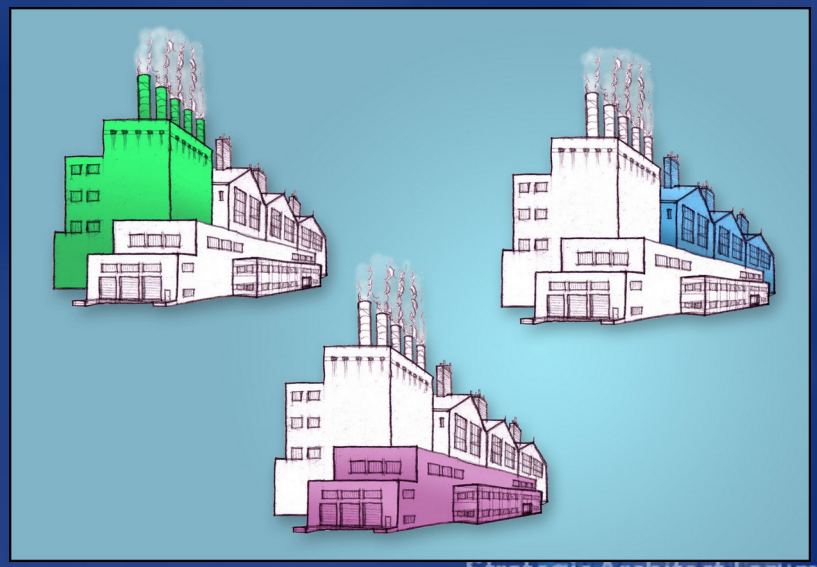
- Business Function Outsourcing
  - Acquire Business Functionality Outside
  - Produce More Sophisticated Business Value
  - Concentrate on Your Center of Excellence
- Competition Drives Efficiencies
  - Better Quality and Price
- Requires Standards
  - Data and Biz-Process
  - Normal Case Avoids Human Touch
  - Special Cases Need



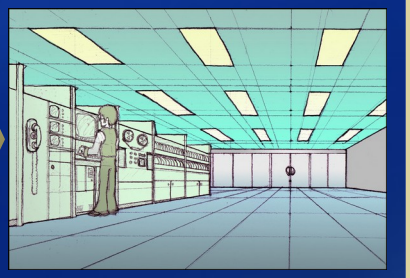
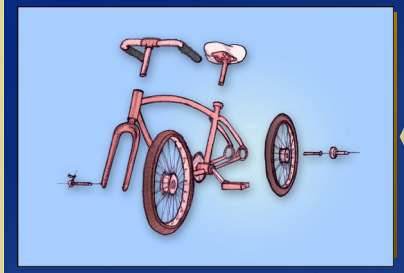
Manufactured Assemblies and Virtual Enterprises

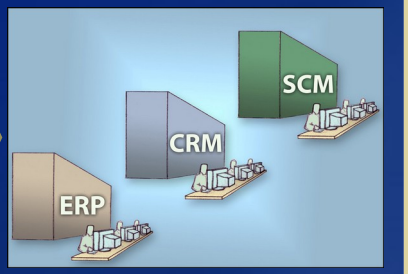
- Standards Allow Composable Stuff (Data)
- Better Stuff (Data) Is Created
  - Combining Efforts of Many Companies
- Increased Efficiencies
  - Lower Prices or Greater Profitability
  - Companies Focus on Specialty





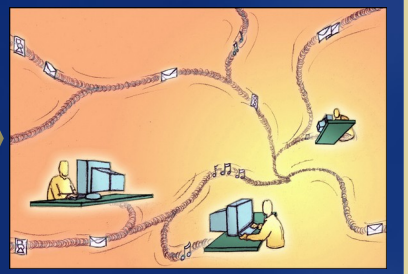
Metropolis

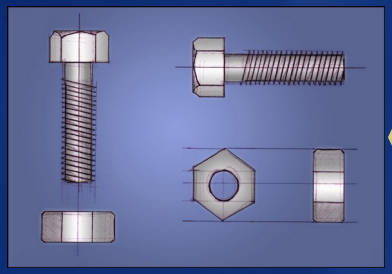
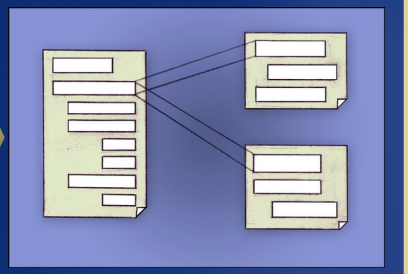
Retail & Distribution

Business Process

Urban Infra-structure

IT Infra-structure

City Government

IT Governance

Retail and Distribution

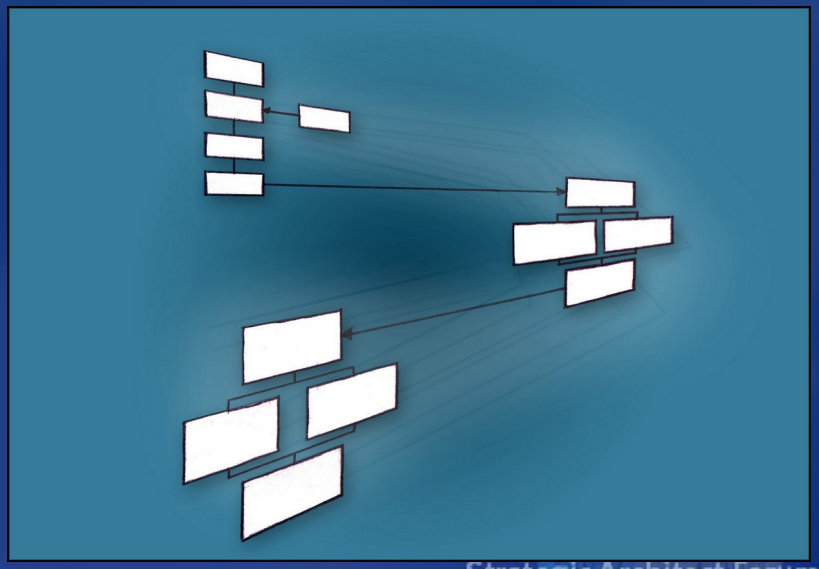
- Late 19th Century
  - Bring People to Stores
    - Trains Brought Shoppers
  - Bring Standardized Stuff to Stores
    - Department Stores Emerge
  - Send Standardized Stuff to People
    - Mail Order
- 20th Century
  - Department Stores and Supermarkets
  - WalMart
    - *Shift in Power to the Retailer!*

Business Process

- Today:

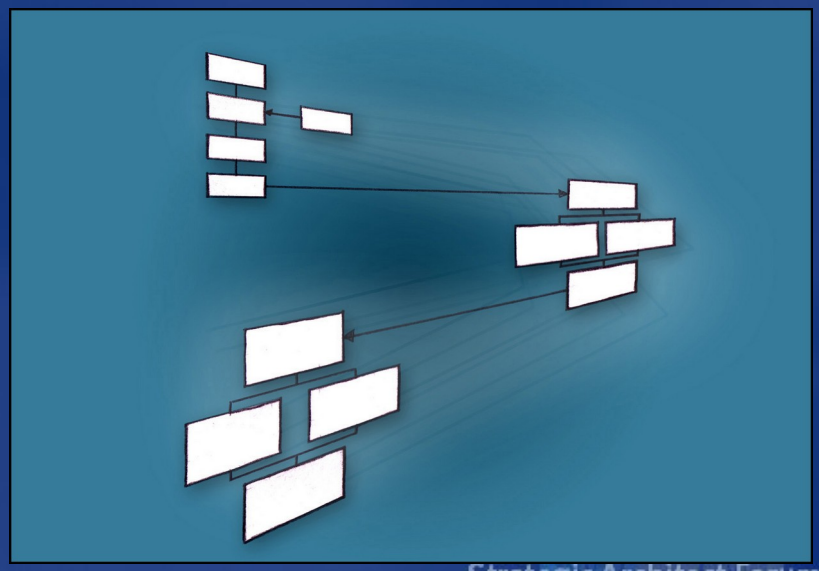
Swivel-Chair Integration	Excellent for EAI across apps
FAX and Pray Integration	Most common form of B2B work
ALT-TAB Integration	Reduces errors via the clipboard

- Looking Ahead:
  - Data and Operation Standards
  - Need “Interchangeability”
    - Standardized Clothes Sizes
    - Not As Specific and Detailed
  - Allows Pre-Allocation
  - Makes Biz Process Efficient

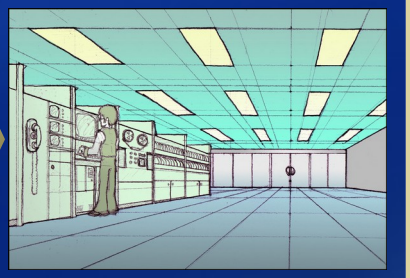
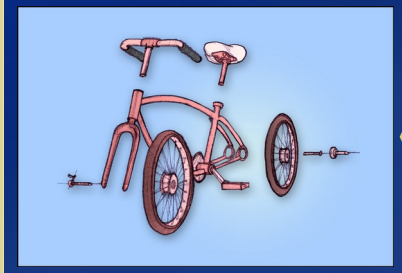
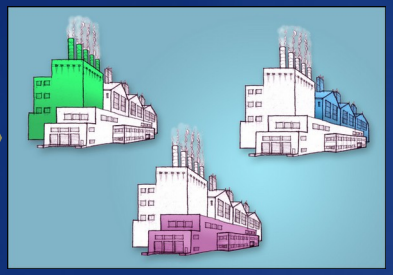


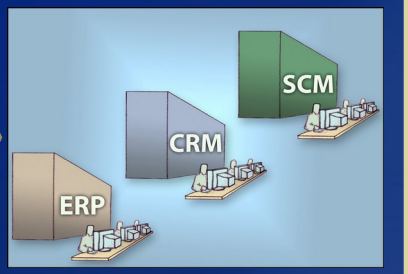
Retail and Business Process

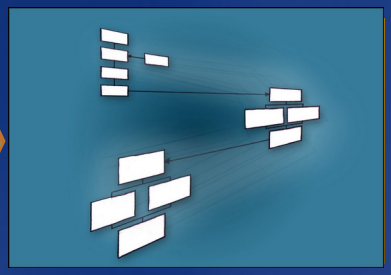
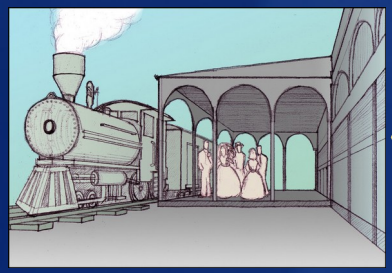
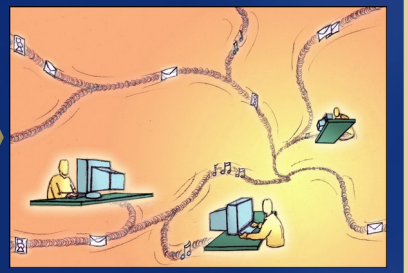
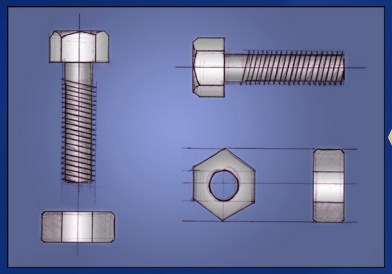
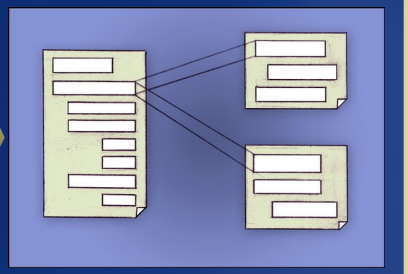
- Amazing Transformation in Retail
  - People Cheerfully Accept Standard Stuff
    - Customization Is Rare and Expensive
- Business Process Mostly Hand-Crafted
  - Poor Standards; Manual “Trim and Shim”
  - Poor “Interchangeability”
- Business Process Will Grow to Drive the Apps!



Metropolis



Urban Infra-structure

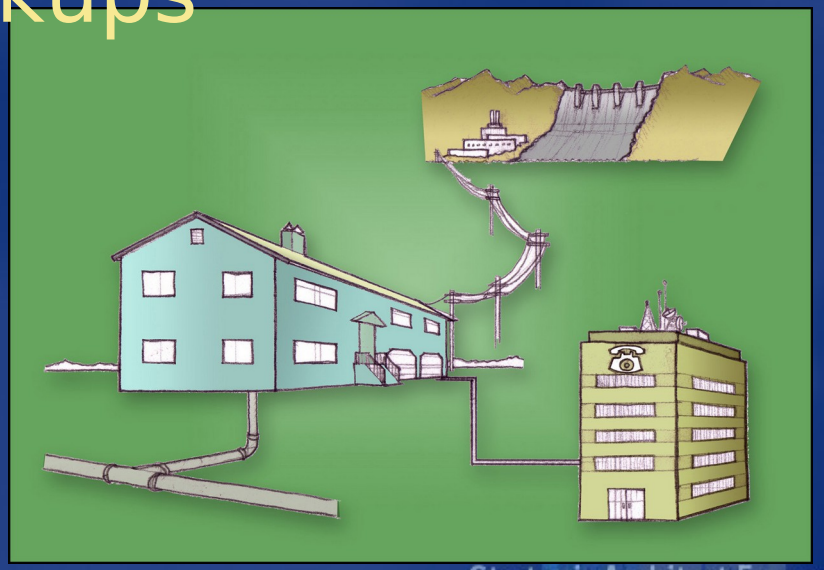
IT Infra-structure

City Government

IT Governance

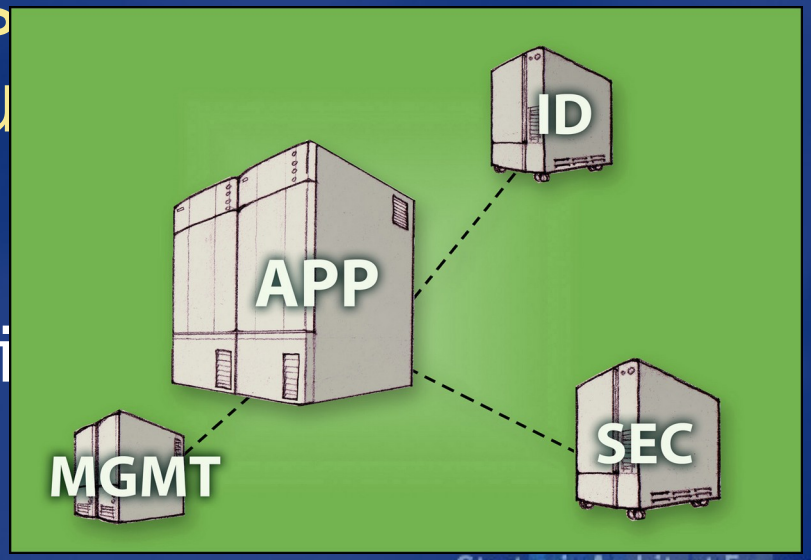
Urban Infrastructure

- Urban Density → Urban Infrastructure
  - Water, Sewer, Gas, Electricity, Broadband…
- Requires Metropolitan Support
- Requires Local Hookups
- Retrofit Happens
  - Notre Dame Has Toilets
  - Conduits for Future
- Funding
  - Sometimes Private



IT Infrastructure

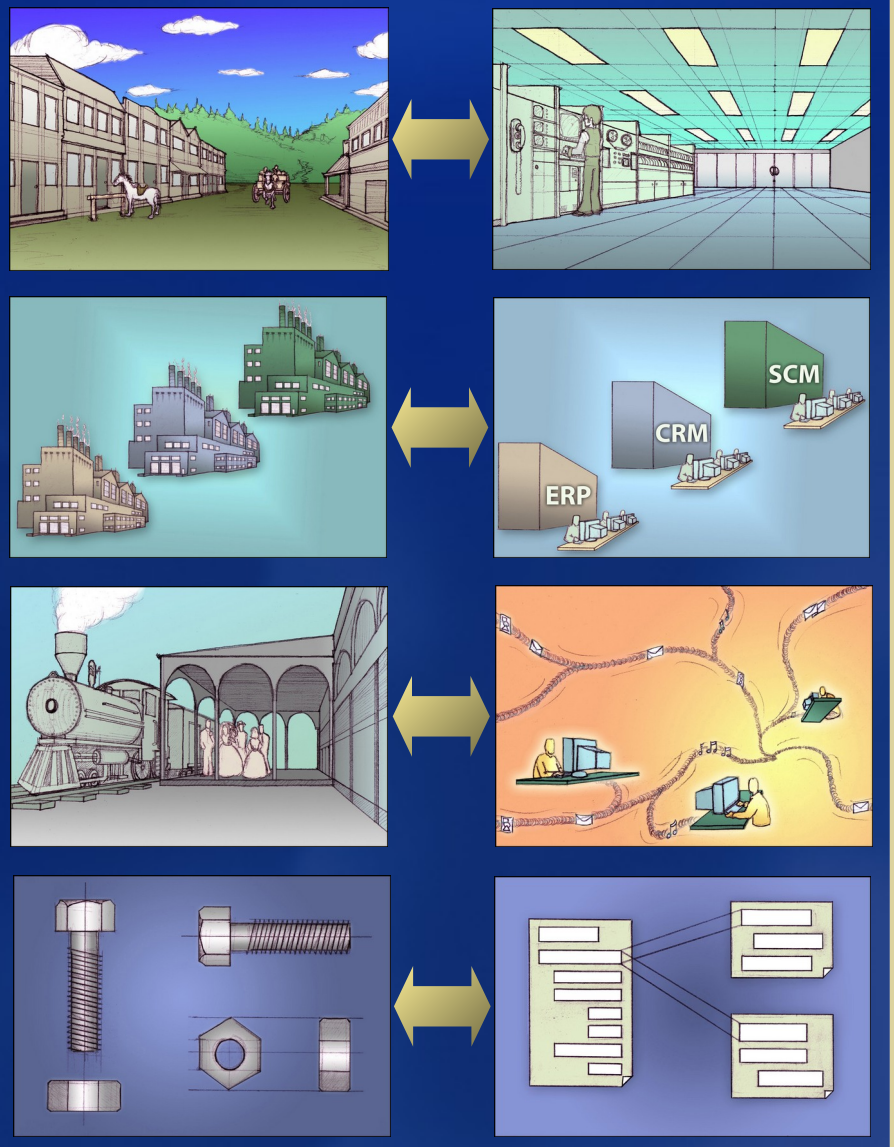
- Many Apps and Servers → IT Infrastructure
  - Easier with Single Mainframe
- Federated Infrastructure:
  - Identity, Security, Naming and Directory, etc.
- Requires IT Services
- Requires App Hooku
- Retrofit Happens
  - Web Services Offer Hope to Ease Retrofi
- Funding Competes with Apps

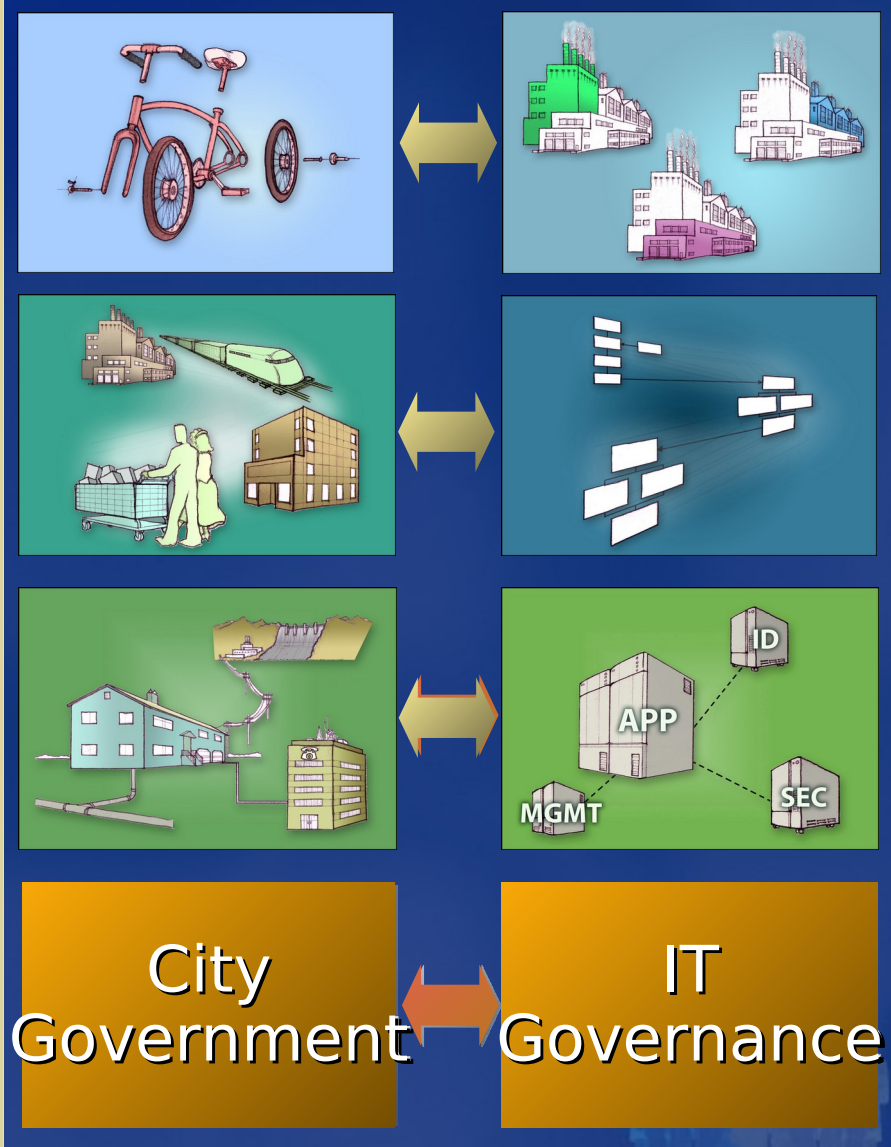


Urban and IT ure

- Infrastructure for Crowded Environments
  - Needs Supporting Services
  - Needs Hookup to Buildings or Apps
- Retrofit Happens
  - May Be Biggest Cost
  - Design for Future Extensions
- Funding Competition

Metropolis





City Government

IT Governance

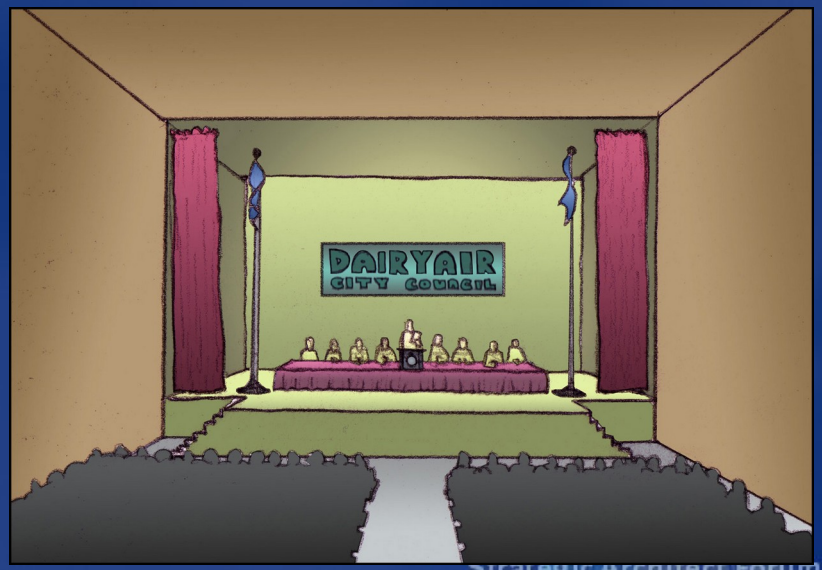
City and IT Decisions

City Vision	IT Principles
City Planning	IT Architecture
City Infrastructure Goals	IT Infrastructure Strategy
Factory/Building Investments	Business Application Needs
Private/Public Investment Choices	IT Investment and Prioritization

Source for IT Decisions:
"Don't Just Lead, Govern! Governing IT for Different Performance Goals"
Professor Peter Weill, CISR, MIT Sloan School of Management; June 2003

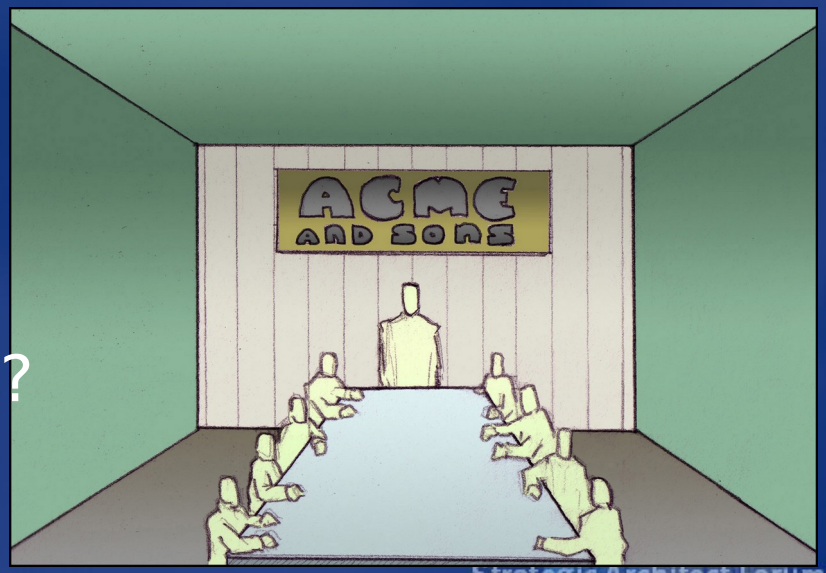
City Government

- Decisions for Allocating Resources
  - City and Business Leaders
  - Infrastructure Usually Needs Cooperation
- Factories/Buildings Usually Business-Driven
  - Cities Usually Constrain and Control
- Manufactured Goods Controlled by Business
  - Cities Have Little Say…

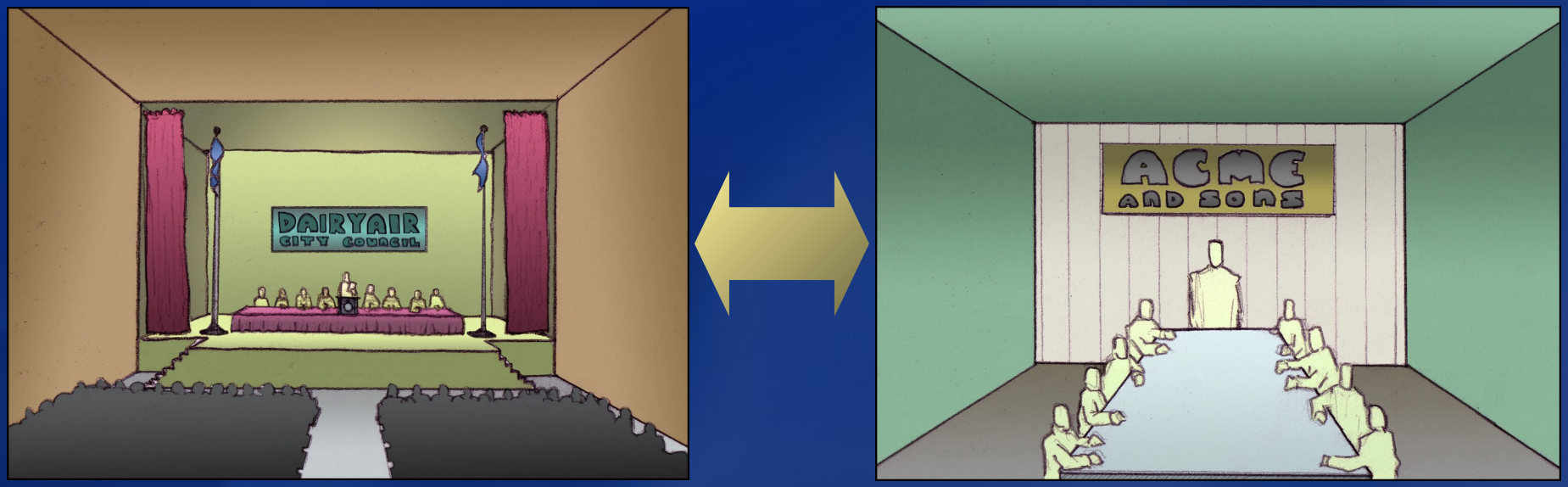


IT Governance

- Who Makes Decisions?
  - CEO? CIO? Business Unit Execs? Techies? Committees?
- What Are Priorities?
  - Asset Utilization? Cost? Flexibility? Growth?
- Metrics
  - What Is Success?
    Who Is Accountable?
- What Are Our Goals?
  - Reduce Cost?
  - Better Information?
  - Competitive Advantage?



City Government and IT Governance



- Similar Decisions and Problems for Cities and IT Shops
  - Cities Provide Inspiration
- Cities (Usually) Optimize for Growth
  - Businesses Drive Building/Factory Investment

See: "Don't Just Lead, Govern! Governing IT for Differe/E Performance Goals," IBID

Metropolis

Looking to the Future

- Equivalent to 1880 or So…
  - Communication and Browsing Well Established
  - Virtual Enterprises Getting Going
  - Business Process a Gleam in Our Eye
- Lots of Fun Ahead of Us
  - Biz Process Won't Take As Long As Retail Did to Mature

Implications of Metropolis

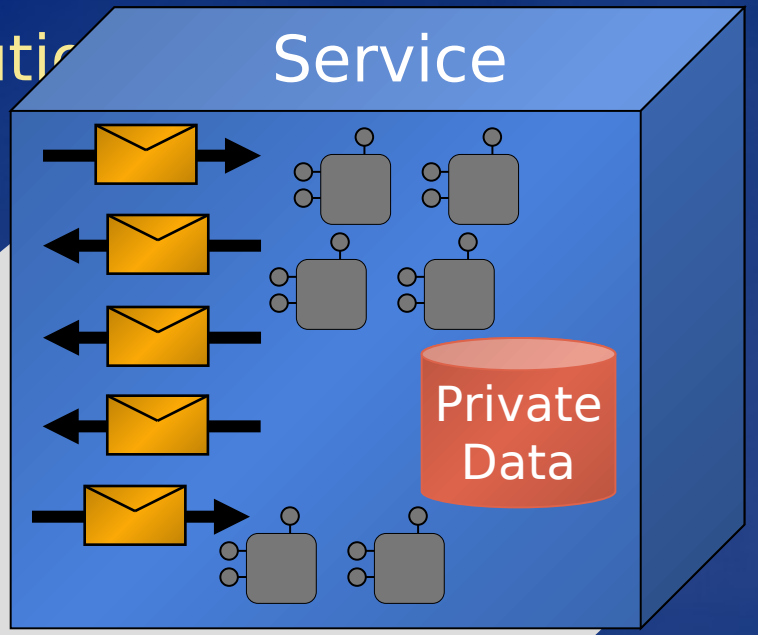
- Heterogeneity Happens!
- Ongoing IT Investment
  - Infrastructure versus Business
  - Historic Monuments
- Standardization Is Nascent
  - Connection Largely by People
  - Efficiencies Still to Come
- Business Process Is Nascent
  - Still Mostly Ad-hoc
  - Growing to Become Dominant Force
- Loose Coupling Helps Investments

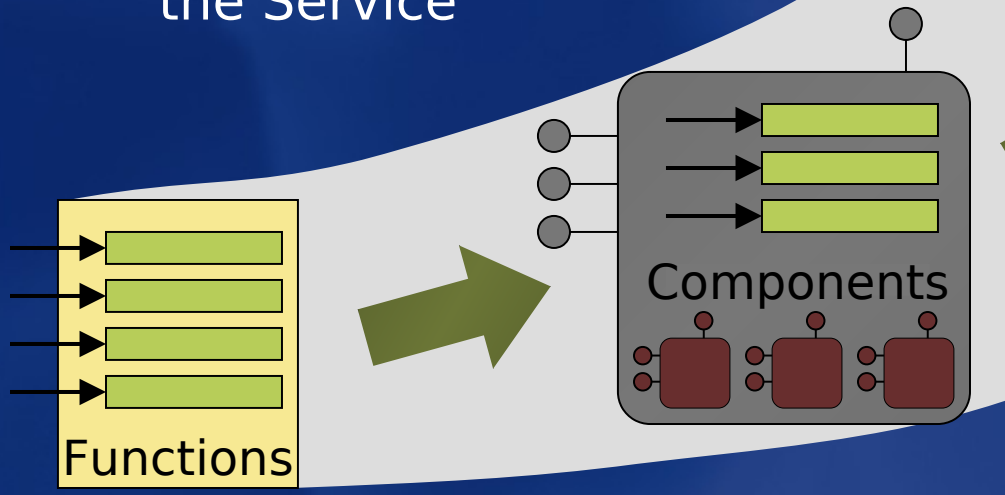
Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - **Introduction**
    - Avoiding Ambiguity in Messages
    - Services for the Enterprise
    - Message Delivery in the Mean, Cruel World
    - Implementing Your Business Service
    - Embracing and Extending Your Existing App
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

Services vs. Components

- There Has Been a Natural Evolutio
  - Function → Component → Service
- Services Comprise Messages, Components, and Data
  - The Private Data Is Completely Encapsulated by the Service
  - Messages Are the Only Ways into and Out of the Service





Services allow for a loosely coupled relationship; Components are much more intimate.

Services and the Enterprise

- Ubiquitous Services
  - Lots of Work on Services As a General Mechanism
    - Services in the Small and Services in the Large
- "Longhorn," Services, and Web Services
  - Services Will Enhance the Programming Model
  - Tools and Frameworks Will Make This Easier
  - Web Services Offer Standardized Interoperability
- Zooming Out to the Enterprise
  - We Will Talk About the Enterprise Usage of Services
  - More Prescriptive (Yet Compatible)
- Platform-Independent Architecture Discussion
  - Deliberately Focusing on Concepts and Techniques
  - Applicable Across Many Platforms

Challenges with a Service-Oriented Enterprise

- Apps Have Developed Independently
  - Purchased and/or Home-Grown
- Usually, Apps Built to Work with Humans
  - Not Designed to Work with Apps
- We Need Services and We Need to…
  - Surround Existing Apps to Play As Services

What're We Gonna Talk About?

Practical Advice for Building Your Services *Now* !

Avoiding Ambiguity in Messages	What to consider when writing messages to ens they are understood by your partner
Services for the Enterprise	Special considerations for using services for enterprise-class applications
Message Delivery in the Mean, Cruel Worlc	Connecting apps (services) with messages has many pitfalls and failure windows... What to do?
Implementing Your Business Service	Messages, data, and transactions... How can we make an enterprise-class service work?
Embracing and Extending Your Apps	I've got a LOT of existing applications... How ca they be tied into the services game?

Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - Introduction
    - **Avoiding Ambiguity in Messages**
    - Services for the Enterprise
    - Message Delivery in the Mean, Cruel World
    - Implementing Your Business Service
    - Embracing and Extending Your Existing App
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

Messages Are Special

- Between Services Is Special
  - Messages Are Sent and Float Around Between
- Special Care Is Needed to Understand Them
  - Can Be Confusion [illegible]terpreting Them

Immutable and/or Versioned Data

- Data May Be Immutable
  - Once Written, It Is Unchangeab
- Immutable Data Needs an ID
  - From the ID, Comes the Same Data
  - No Matter When, No Matter Where
- Versions Are Immutable
  - Each New Version Is Identified
  - Given the Identifier, the Same Data Comes
- Version-Independent Identifiers
  - Let You Ask for a Recent Version

Microsoft Windows NT® 4.0, S
- The Same Set of Bits Every Time

Recent NY Times
- Maybe Today's, Maybe Yesterday's

Version-Independent

New York Times; 7/3/03
- Specific Version of the Paper
  – Contents Don't Change

Latest SP of NT 4.0
- Definitely Windows NT 4.0, Results Vary over Time

Stability of Data and Metadata

- Immutability Isn't Enough!
  - We Need a Common Understanding
  - President Bush → 1990 vs. President Bush → 2003
- Stable Data Has a Clearly Understood Meaning
  - The Schema Must Be Clearly Understood
  - [illegible]etation of [illegible]us

Suggestion
- Timestamping or Versioning Makes Stable Data

Observation
- A Monthly Bank Statement Is Stable Data

Advice
- Don't Recycle Customer IDs

Observation
- Anything Called "Current" Is Not Stable

Normalization and Read-Only Data

- Databases Design for Normalized Data
  - Can Be Changed Without "Funny Behavior"
  - Each Data Item Lives in One Place
- Sometimes Data Should Be De-Norm
  - If Data is Read-Only

De-normalization is OK if you aren't going to update!

Classic problem with de-normalization:

Can't update Sam's phone # since there are many copies

Emp #	Emp Name	Emp Phone	Mgr #	Mgr Name	Mgr Phone
47	Joe	5-1234	13	Sam	6-9876
18	Sally	3-3123	38	Harry	5-6782
91	Pete	2-1112	13	Sam	6-9876
66	Mary	5-7349	02	Betty	4-0101

To Cache or Not to Cache?

- OK to Cache Immutable Data
  - It's Never Wrong
  - Never Have to Shoot Down the Cache!
- Consider Stability of the Data
  - Don't Want to Misunderstand It
- Messages Are Easily Cached
  - They Should Be Immutable and Stable
- Reference Data Should Be Versioned
  - The Version Is Immutable
- Some Data Should Not Be Cached
  - If It Changes, the Cache May Be out of Sync

Validity of Data in Bounded Space and Time

- Bounding the Valid Times
  - It May Have an Expiration
- Bounding the Valid Locations
  - Restrictions on Where the Data Is Valid
- When Valid, the Data Should Be:
  - Immutable (the ID Yields the Same Bits)
  - Stable (the Meaning Is Clea

Price List Valid Until Dec 31st

Data Valid for Service X Only

"Offer Good Until Next Tuesday"

"Offer Good to Washington State Residents Only"

Rules for Sending Data in Messages

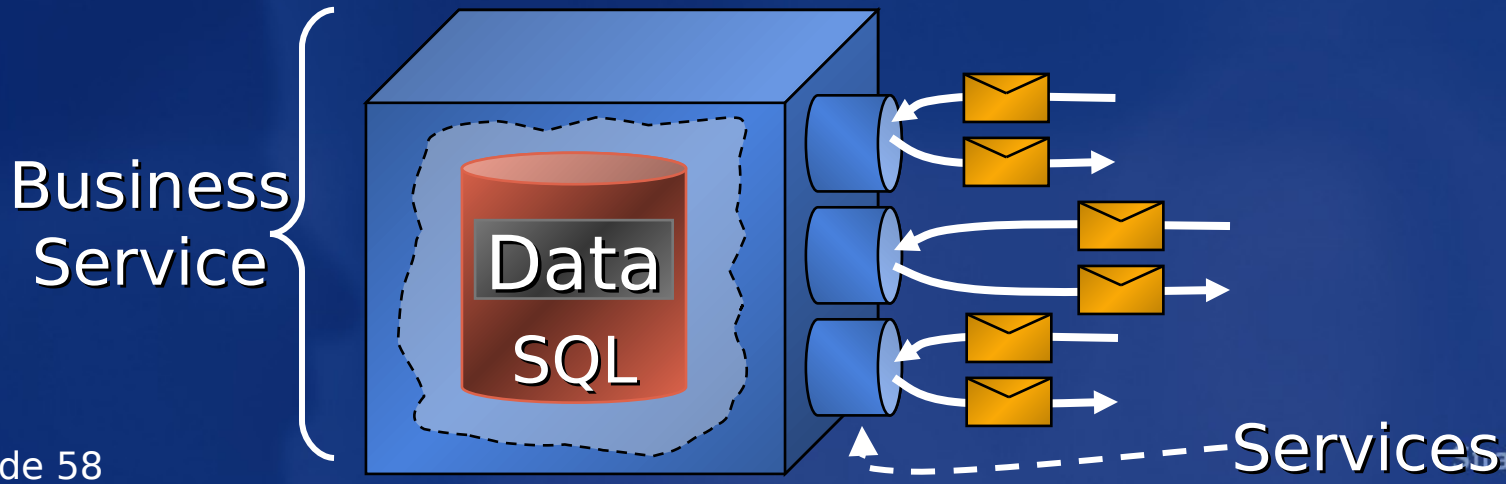
Identify the Message	Put Unique ID in All Messages Part of the Unique ID May Be a Version…
Immutable Data	Don’t Change the Data Associated with the Unique ID; Never Return Different Bit
OK to Cache	The Same Bits Will Always Be Returned
Define Valid Ranges	Valid for a Certain Time Period and over Some Space; OK to Always Be Valid
Must Be Stable	Must Ensure There Is Never Any Confusic About the Meaning (Within Valid Range)

Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - Introduction
    - Avoiding Ambiguity in Messages
    - **Services for the Enterprise**
    - Message Delivery in the Mean, Cruel World
    - Implementing Your Business Service
    - Embracing and Extending Your Existing App
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

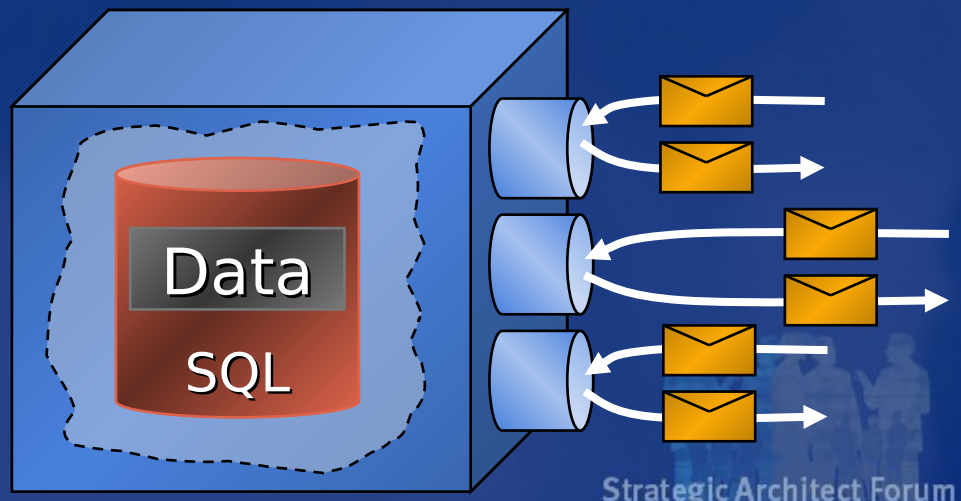
Services and Business Services

- Services Provide a Name for Messaging
  - Messages Target the Service
- Enterprises Need Business Services
  - Projects One or More Services to the Outside World
  - Encapsulates Some Database-Resident Data
  - Implemented with Business Logic
- Business Services Are a Prescriptive Usage for Services
  - Comply with the General Mechanism
  - OK to Use Web Services for Interoperability
  - OK to Use Some Other Mutually Agreed Protocol



Complete Encapsulation of Data in a Business Service

- Business Services Encapsulate Their Data
  - The Only Way to the Data Is via Messaging
- Messages Contain:
  - Operation Requests
    - Perform a Business Function
  - Operation Responses
    - Describe the Outcome and Provide Any Results
  - Processed Data
    - Carefully Designed for Outside Use
- Sometimes Export for Data Warehousing
  - Always to a Trusted Partner Node



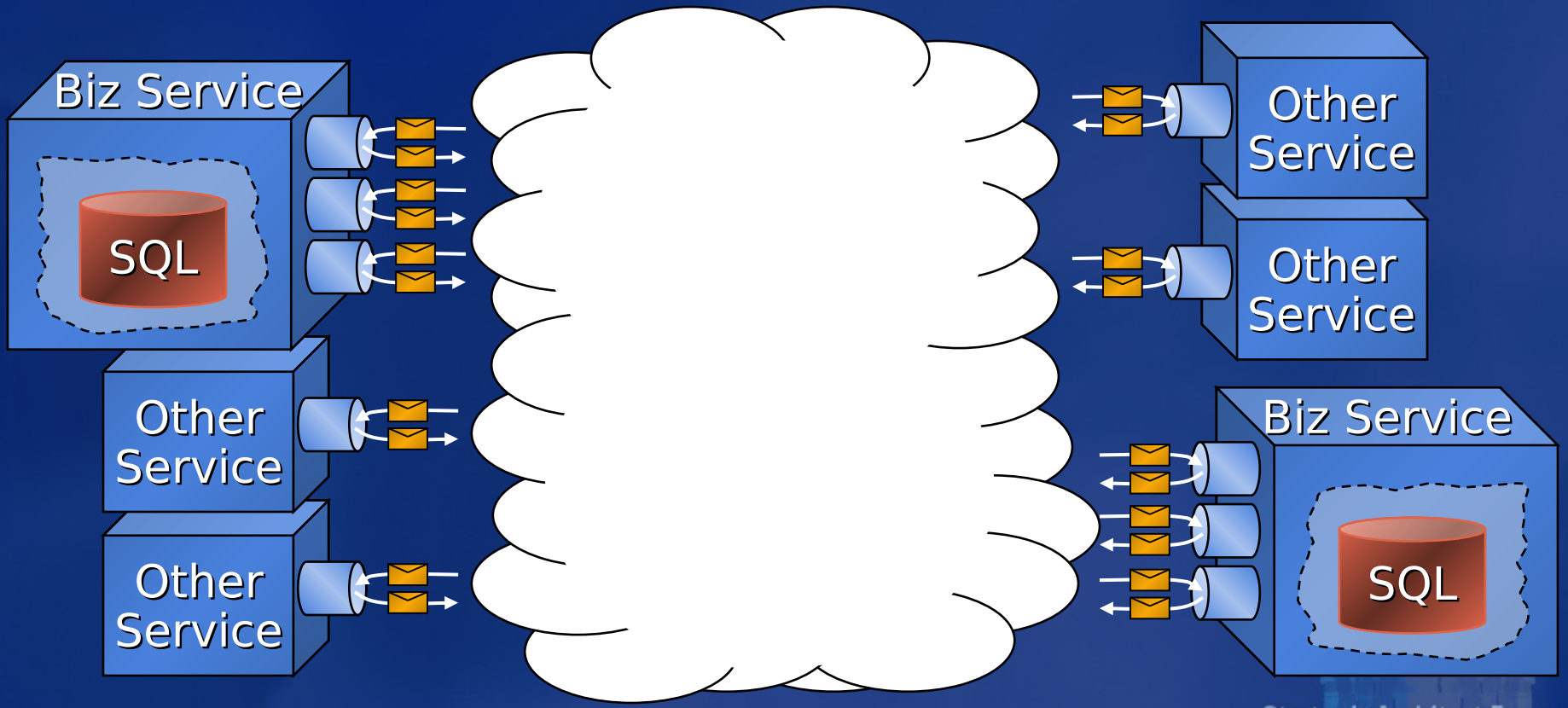
The data is completely encapsulated within the business service.

Storing Data That Is Encapsulated in a Business Service

- A Business Service Should Use SQL to Store Data
  - Many Advantages for Enterprise Use
- Data Lives in a Set of Partitions of a Set of Tables
  - Set of Tables
    - Business Service Implements a Set of Services
    - These Services Each Use a Set of Tables
  - Set of Partitions
    - Sometimes a Business Service Is Partitioned
- Each Business Service's Data Should Live in a Single Database Engine
  - If You Can't Get It into One Engine, Make Two Business Services

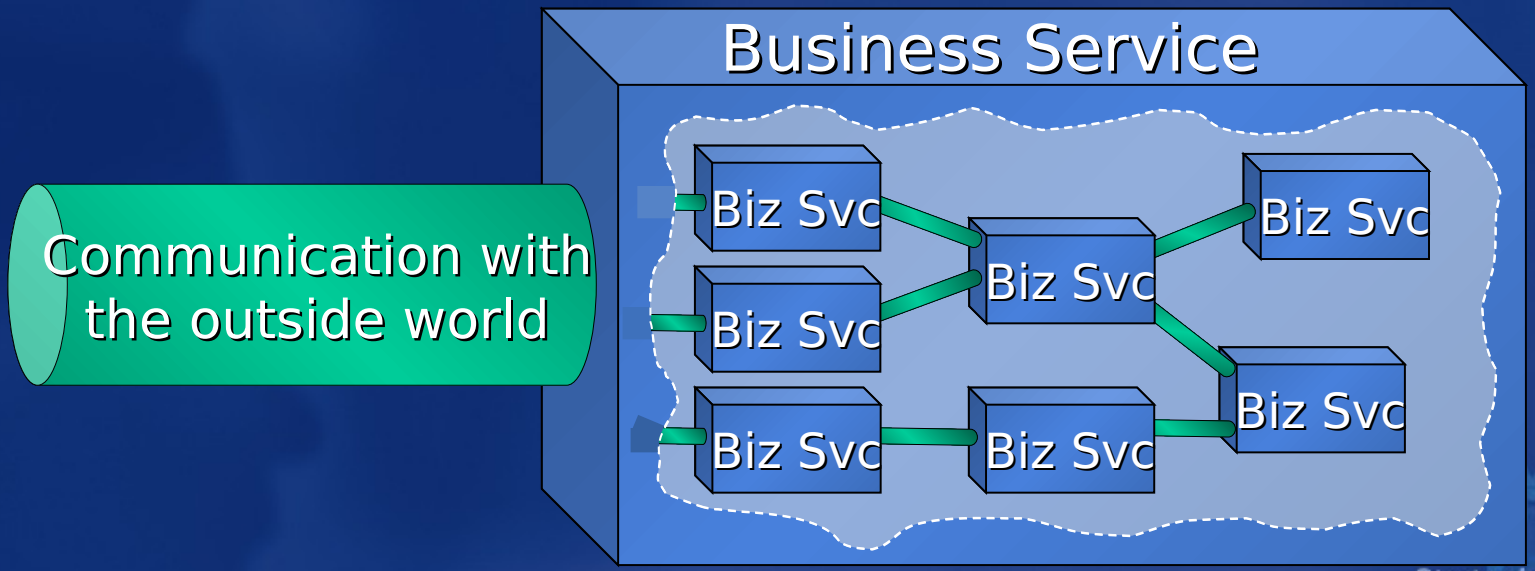
Business Services Are Services

- Business Service IS a Service
  - From the Outside, You Can't Tell the Difference
  - It Is About Design and Implementation!



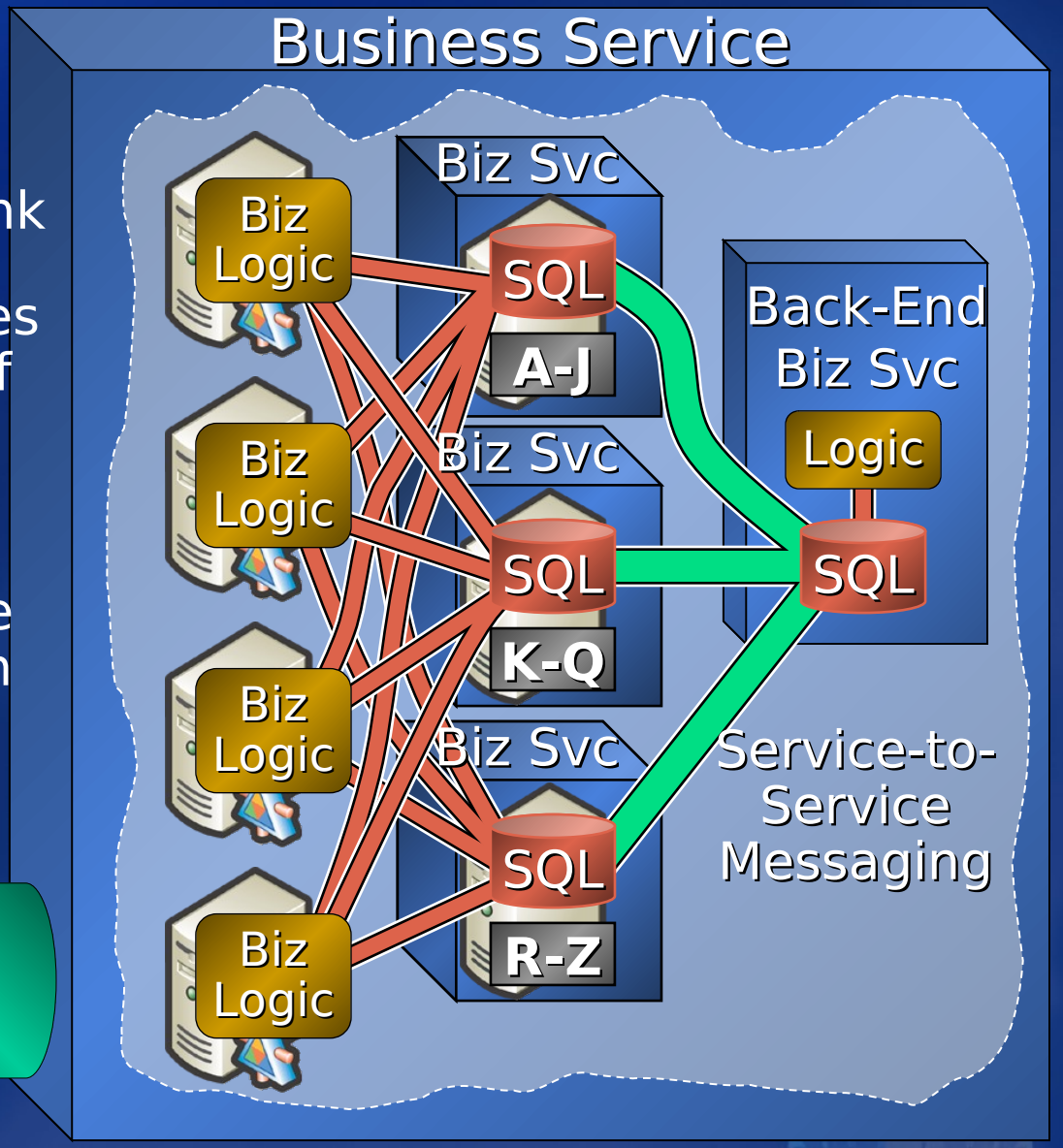
Composite Business Services

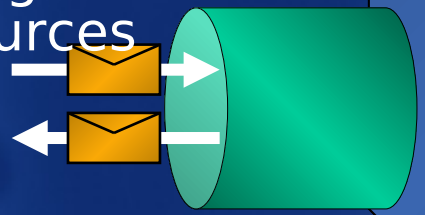
- A Business Service May Be Implemented by Many Business Services
  - Perhaps on Many Machines
  - Essential for Scale-Out Solutions
- Must Look Like a Single Business Service
  - Can't Tell from the Outside



Composite Services and Scale Out

- Scale-Out Trick:
  - Load-Balanced Bank of Compute Engines
  - Partitioned Bank of Database Engines
- Multi-Message Conversation
  - Stored in Database
  - Find Right Partition
- Centralized Back-End Service
  - Manages Shared Resources





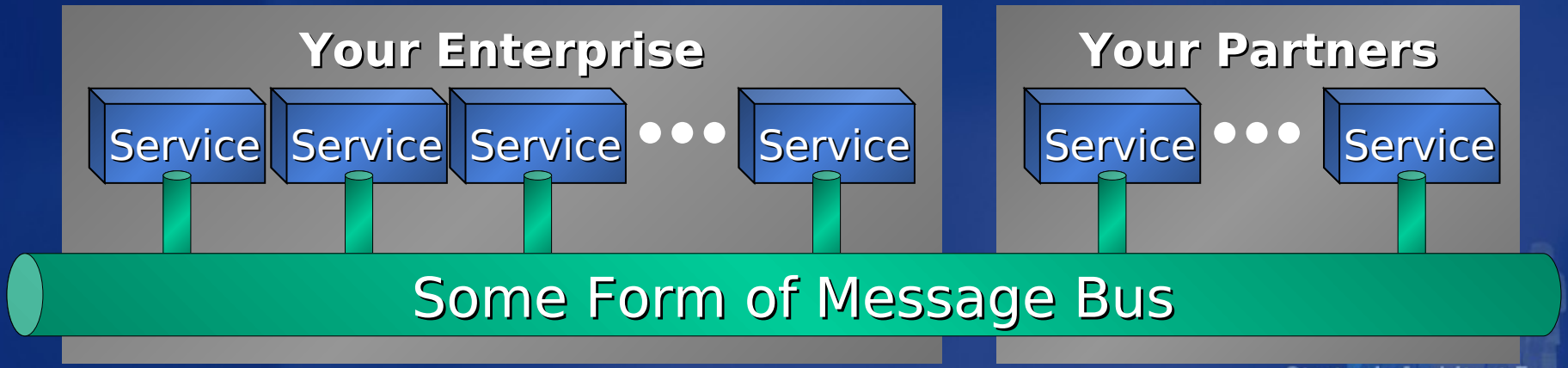
The soul of the biz service is the SQL database!

Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - Introduction
    - Avoiding Ambiguity in Messages
    - Services for the Enterprise
    - **Message Delivery in the Mean, Cruel World**
    - Implementing Your Business Service
    - Embracing and Extending Your Existing App
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

Any Form of Messaging Is OK

- Many Kinds of Messaging
  - E-Mail, IP, TCP/IP, HTTP, MSMQ, MQ Series, and More
- Advantages and Disadvantages to Each
  - HTTP Is Ubiquitous (Huge Advantage)
  - MQ Series—Lots of Platforms
  - E-Mail (over SMTP) Is Ubiquitous (Huge Advantage)
- Ubiquity Is Essential
  - Connecting Within Your Enterprise
  - Connecting with Your Partners
- OK to Have Some Message Loss
  - We Will Discuss How to Cope with This



What Is Guaranteed Message Delivery?

- Guaranteed Message Delivery
  - The Message Gets to Queue or Sender Is Told
  - MSMQ, MQ Series, and Others…
- Need to Know the Service Processed Message
  - And Was the Processing Successful?
- Need to Know the Response to the Message
  - The Service Got the Message
  - The Service Processed the Message
  - What Was the Outcome?
- Knowing It's in the Queue Isn't Enough…

Idempotence

- Requests Get Lost…
  - Gotta Retry Them to Handle Lost Requests
- Requests Arrive More Than Once…
  - Those Pesky Retries May Actually Arrive
- Idempotent Means It's OK to Arrive Multiple Times
  - As Long As the Request Is Processed at Least Once, the Correct Stuff Occurs
- In Today's Internet, You Must Design Your Requests to Be Idempotent

Not Idempotent
Withdrawing $1 Billion

Naturally Idempotent
Sweeping the Floor

Idempotent
Baking a Cake Starting from the Shopping List (If Money Doesn't Matter)

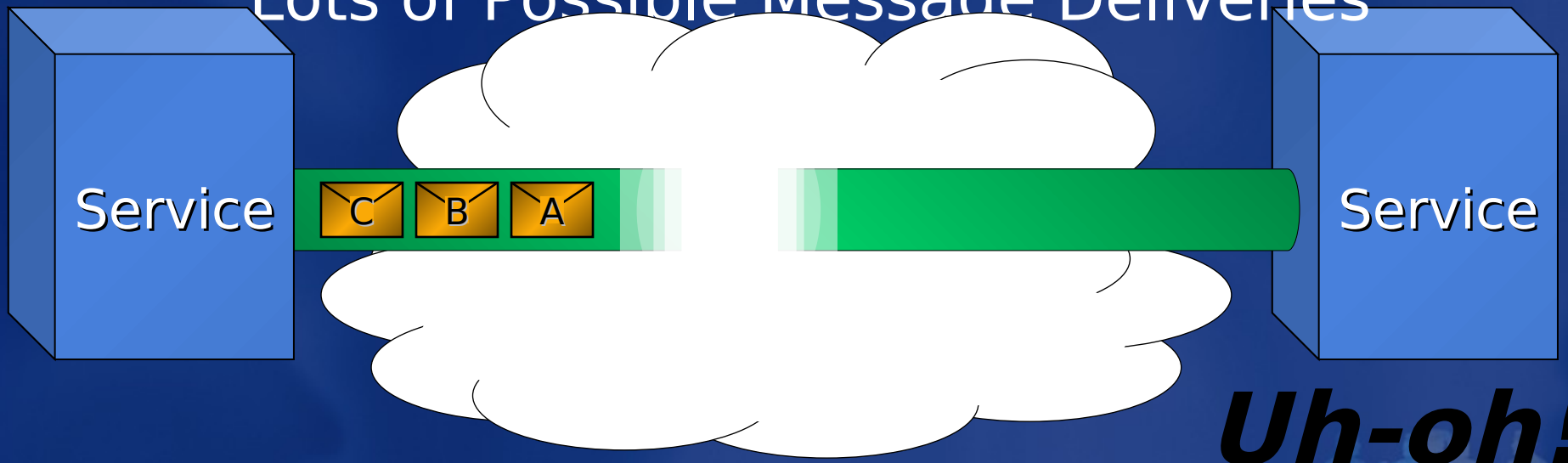
Not Idempotent
Baking a Cake Starting from Ingredients

Idempotent
If Haven't Yet Done Withdrawal #XYZ for $1 Billion, Then Withdraw $1 Billion and Label As #XYZ

Naturally Idempotent
Read Record "X"

Challenges with Multiple Messages

- Any Message May Arrive Multiple Times
  - Even After a Long While
- This Can Be Very Confusing…
  - Lots of Possible Message Deliveries



Uh-oh!

Using Request Response with Idempotence

- You Can Ensure the Processing Occurs in Order!
  - Make Each Request Idempotent
    - Retries Won't Matter…
  - Use Request/Response Pairs
    - Each Request Submitted Has a Matching Response
  - Implement Timeout and Retry
    - If You Don't Get a Response, After a While, Try Again
- Request/Response with Idempotence Works!
  - It Works Even over Flakey Messaging
    - The Response Comes from the Service Itself
    - Much Better Than Guaranteed Messaging
  - Request/Response Can Be Constraining
  - Idempotence Can Be a Difficulty
    - Discussion Below About How to Ease That Pain

Statefulness and Multi-Message Interactions

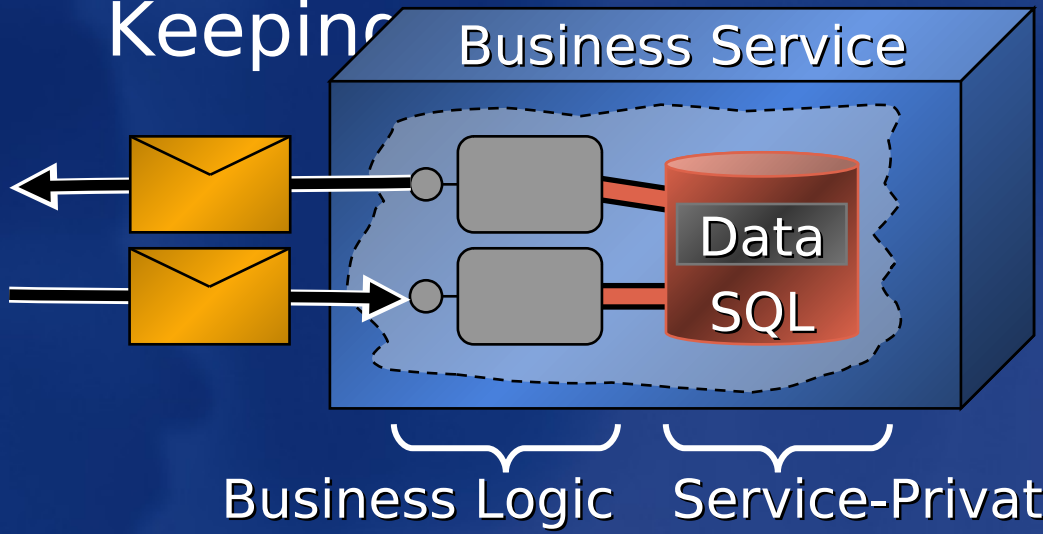
- What If a Message Is Related to an Early Message?
  - How Do We Associate Them?
  - How Do We Remember Data Across Them?
- A Conversation Is a Set of Related Messages
  - Requests Flowing One Way; Responses the Other
- Consider Request/Response Pairs
  - Richer Patterns Are Harder
- There May Be Many On-Going Conversations
  - Each Must Be Kept Separate in the Database

Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - Introduction
    - Avoiding Ambiguity in Messages
    - Services for the Enterprise
    - Message Delivery in the Mean, Cruel World
    - **Implementing Your Business Service**
    - Embracing and Extending Your Existing App
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

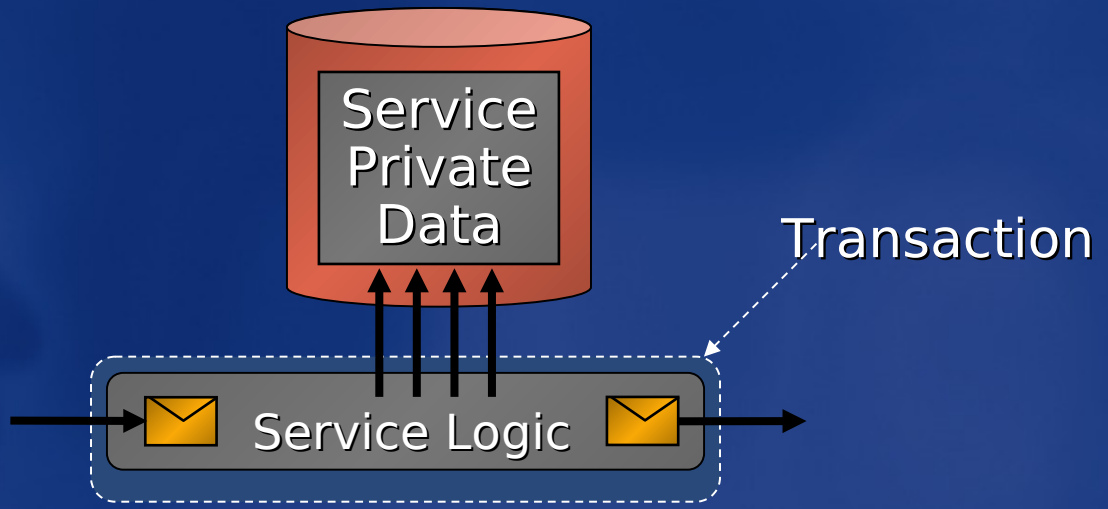
Service-Private Data

- Each Business Service Has Service-Private Data
  - Lives Deep Inside
  - Surrounded and Protected by Business Logic
  - Kept in a SQL Database for Safe Keeping



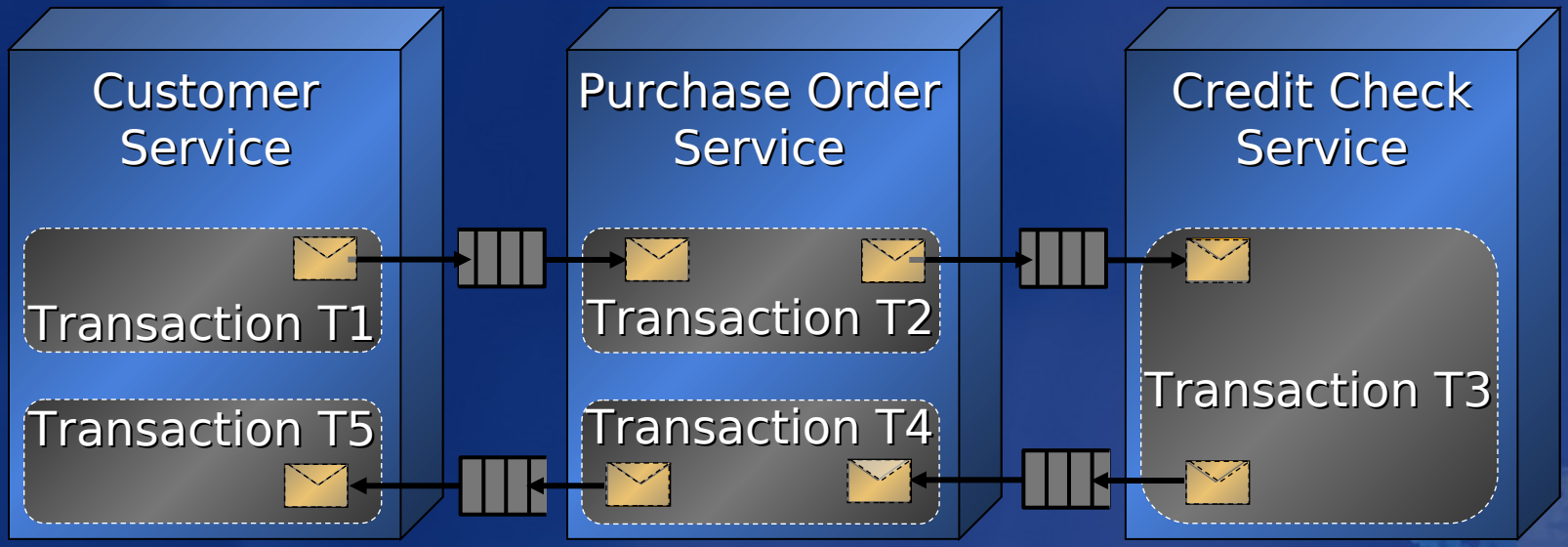
Business Logic Surrounds Service-Private Data

- Business Logic
  - Surrounds and Protects the Private Data
  - Only Changes Approved by Business Logic
- Messages Stimulate Work
  - Logic Changes Service-Private Data
  - Generates Outgoing Messages
  - Atomic Transaction



Transactions and Multiple Services

- Distributed Transactions Have Challenges
  - Performance and Autonomy
- Long-Running Business Operations
  - Multiple Transactions over Different Business Services
  - Multiple Transactions over Same Business Service
  - Connected by Messaging



Read-Only Requests

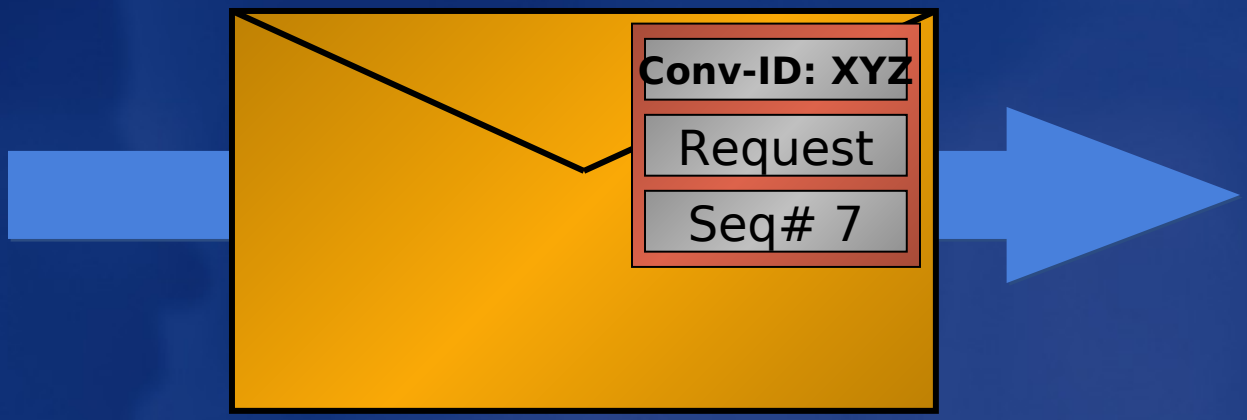
- Read-Only Requests Are Idempotent
  - Read and Re-Read… No Problem
  - Any of the Answers Are Just Fine
- OK to Not Record Read-Only Requests
  - (Arguably) a Big Performance Gain
- Most Businesses Want to Log Read Requests
  - Valuable Data for Analysis
  - Can Use Initial Read to Set Up Conversation

Building a Conversation for Multiple Messages

- Conversations Help with Correctness
  - Can Ensure Automatic Idempotence
    - Even When Changing Data
    - Not for First Message in Conversation
  - Correlate Messages and Data
    - Connecting Them to Earlier Messages
    - Allowing Data to Be Correlated
- Conversations Must Be Long-Running and Durable
  - They Live As Records in the Database
- Conversations Connect Business Services
  - Using the SQL Database As the Conversation Storage

Conversation IDs and Sequence Numbers

- Conversation IDs
  - Any Unique Number (Must Not Recycle)
- Sequence Numbers
  - Incremented for Each Request
- Target Service Only Accepts Next in Sequence
  - A Request Cannot Be Processed Twice
  - The Conversation Provides Idempotence
- Responses Match the Seq# of the Request
  - The Response Cannot Be Processed Twice
  - The Responses Are Idempotent



Remembering the Response for Retries

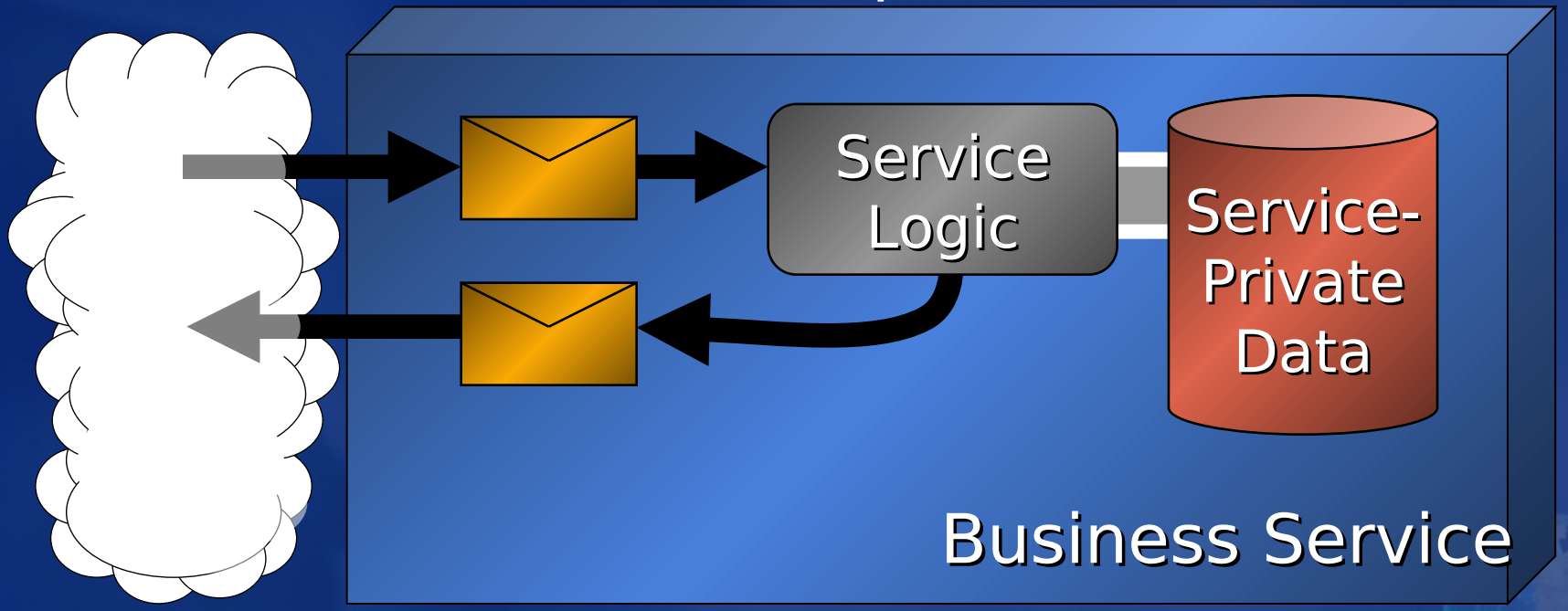
- Retried Requests Sometimes Arrive
  - The Response May Have Been Lost
  - Need to Regenerate the Response
- Don’t Want to Do Request Again
  - Conversations Help with Idempotence
- Remember Responses in the Database
  - If Request Retries, Resend Response

Transactionally Protecting Your Messages

- Processing a Message Should Be Atomic
  - You Mustn't Partially Process the Message
- It's OK to:
  - Do the Work
  - Record the Response
  - Not Send the Response Out (Due to Failure)
- A Retried Request Will Send the Response
- Always Wait to Send the Response Until It Is Recorded
  - Never Send Before Recording

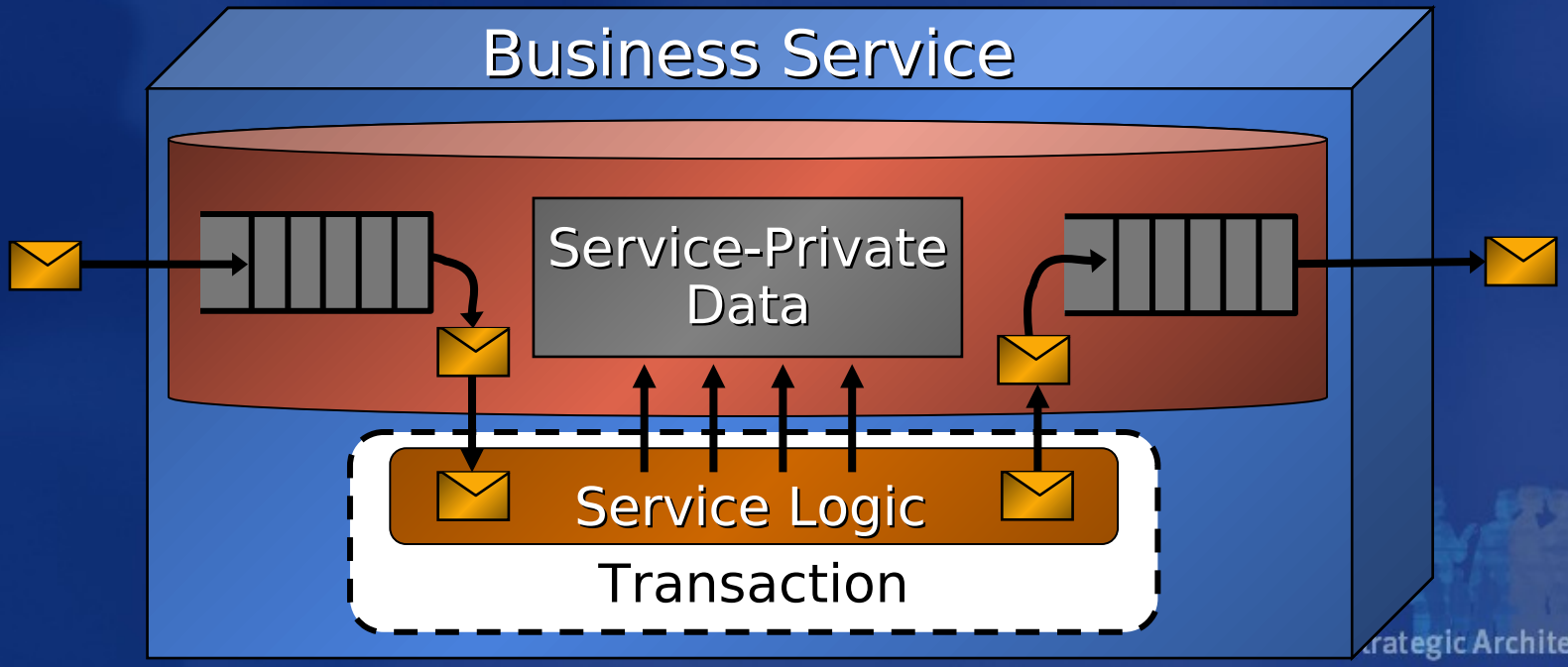
It's All About the Messages, Baby!

- Services Revolve Around Messages
  - Services Are "Black Boxes"
  - Requests Go In; Responses Come Out
  - The Rest Is an Implementation Detail!



It's All About the Data, Baby!

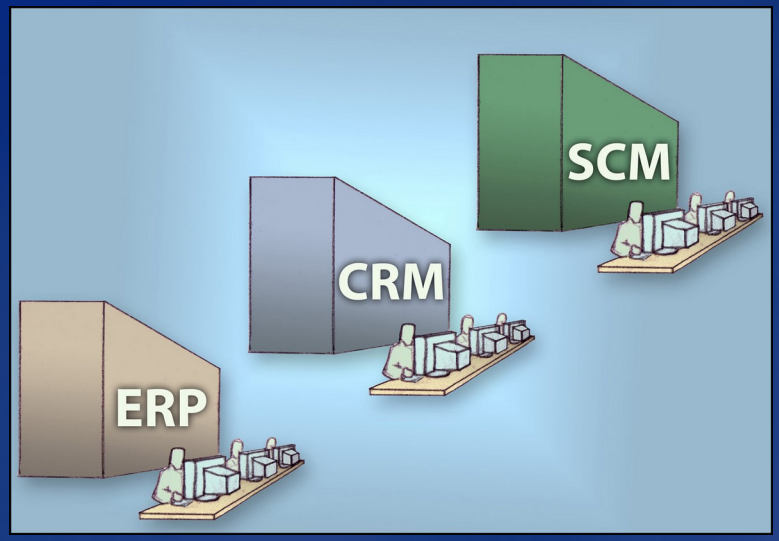
- Messages Are Data!
  - Work Happens by Changing the Database
    - Stimulated by Messages
    - Changes Data
    - Records Outgoing Responses
- A Business Service Live in the Database!



Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - **Practical Advice for Building Services**
    - Introduction
    - Avoiding Ambiguity in Messages
    - Services for the Enterprise
    - Message Delivery in the Mean, Cruel World
    - Implementing Your Business Service
    - **Embracing and Extending Your Existing App**
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

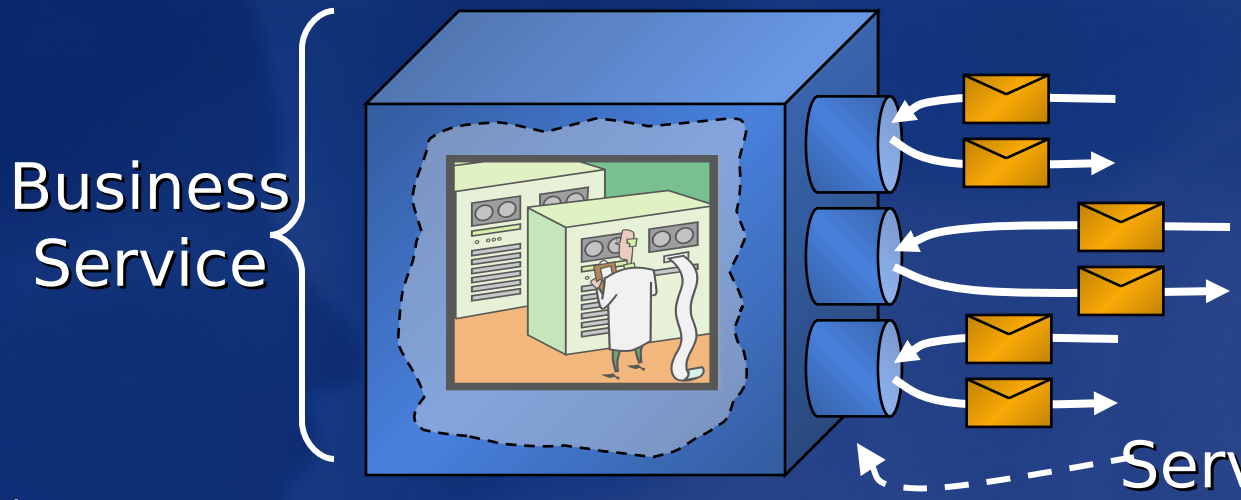
Goals for Wrapping an Existing App



- Most Apps Are Used by Humans
  - People Know How to Make Them Work
  - People Add Judgment to Fit the Circumstances
- *Reduce the Need for People* to Use the App
  - Increase the Times Machine-to-Machine Works OK
  - The Goal Is NOT to Eliminate the Need for People
- Wrap an App to Look Like a Service
  - Increase the Service-to-Service (App) Work

Gaining Entrée to Existing Apps

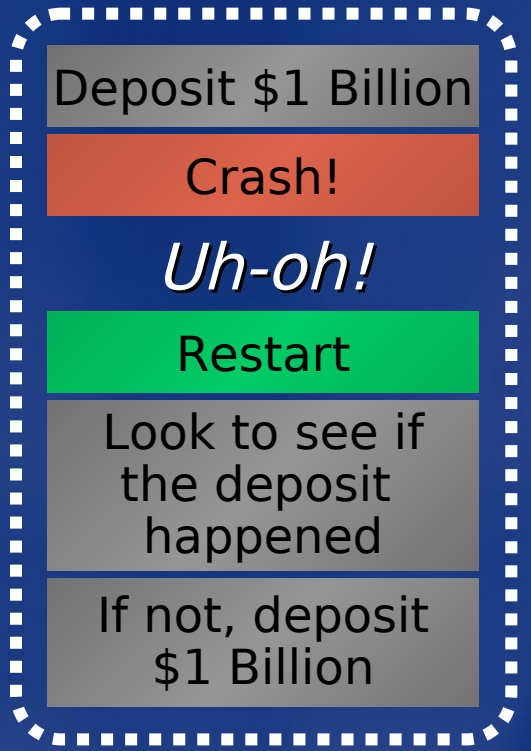
- How Do People Get into the App?
  - HTML, 3270s, 3-Tier, 2-Tier, EDI, MQ, etc…
- Surround and Interface to App
  - Screen Scraping, HTML Parsing, etc…
- Sometimes This Is Hard!
  - Client-Server Needs App Code Changes
    - 2-Tier Especially Hard



If you can tap into the app, you can pretend to be a human for some operations. This can allow the app to look like a service.

Identifying Idempotent Sequences of Operations

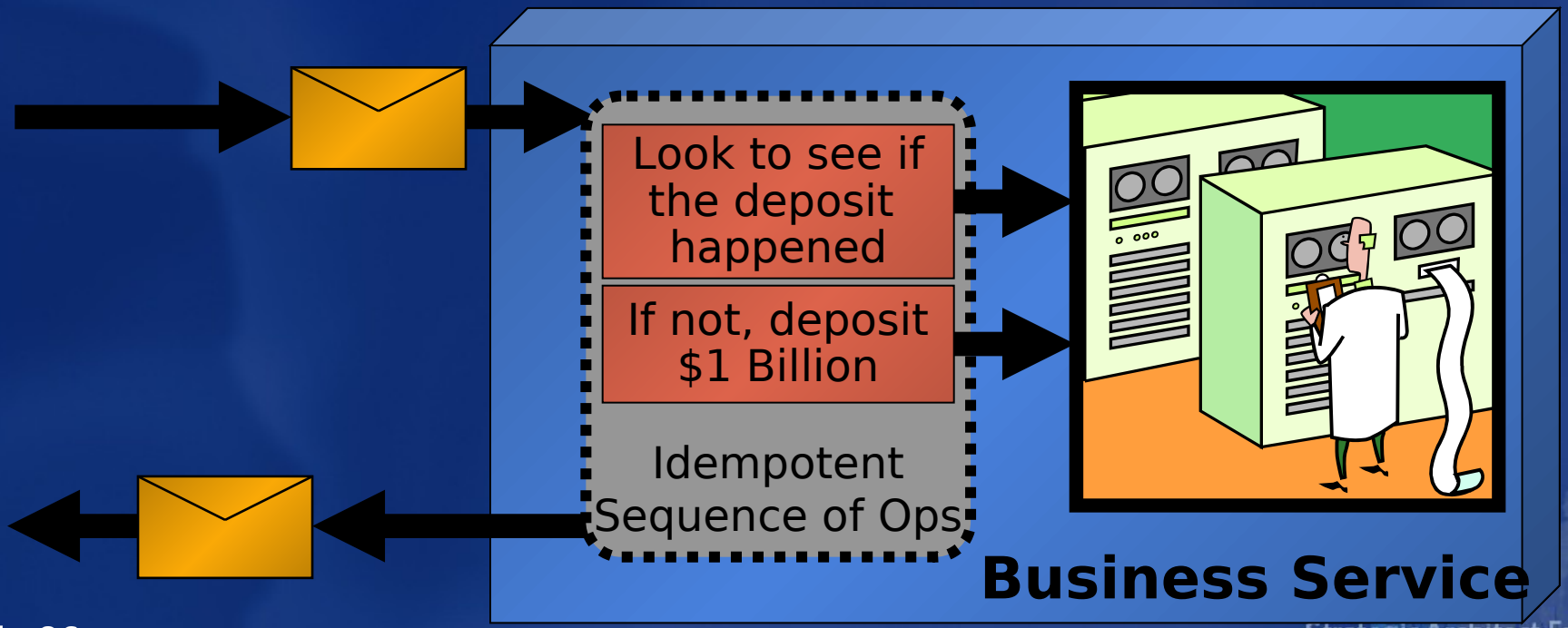
- Every App Deals with Failure
  - People Have Procedures
  - These Are Repeated Until Success
    - Usually Ad-Hoc
- Find the Idempotent Sequences
  - A Set of Operations Against the App
  - Each Separate Op IS NOT Idempotent
  - The Entire Sequence IS Idempotent



Idempotent Sequence of Operations

Defining the Messaging Interfaces

- Define Messages and Sequences
  - Map to Business Operations
  - Search for Idempotence!



Outline

- Metropolis: Part 1
  - Introduction
  - Metropolis: The Analogy
  - Practical Advice for Building Services
  - Concluding Part 1
- Metropolis: Part 2
  - Introduction to Part 2
  - Considering Data and Messaging in Services
  - Thoughts on Business Process
  - Conclusion

Envisioning the Service-Oriented Enterprise

- Services Move Us Forward
  - Lots of Islands of App Code
  - Services Are for Connecting the Islands!
- Independence Is Essential
  - Services Evolve Independently
  - Build versus Buy
- Inside Your Company and Across Companies
  - Connecting Your Disparate Apps Comes First!
  - Hooking to Partners Is Important, Too!
- Services Cleave Your Applications
  - They Cleave Them Apart into Independent Pieces

And Now, a Word From Our Sponsor…

- There's Some Cool Technology Coming!
  - Web Services
    - Defines Interoperability Standards
  - "Indigo"
    - Development Tools and Frameworks for Services
  - "Yukon" SQL Service Broker
    - Integrated Messaging into Databases
    - Automatic Support for Conversational Messaging
    - Transaction-Protected, Reliable Messaging

Part 2: Considering Data and Messaging in Services

The Schema Between the Services	What goes in the messages being passed betwe services? What must be considered?
The Schema Between the Services	What about the different understandings of dat Why is XML such a big deal?
Service Agents and Service Masters	How is data used inside a service? What are the usages patterns that can be leveraged?
Ownership of Data in the Enterprise	If data is changeable, only one service should b in charge… How should this be done?
Representations of Data	XML, SQL, Objects! When do I use what? Why so many choices?
Tying It All Together!	Reiterating the roles for XML, Objects, and SQL

Part 2: Thoughts on Business Process

Interacting with Services	How can we reach agreements without transactions that span services?
Interacting with Masters and Agents	How do tentative operations get applied to Serv Masters? How about Service Agents?
The Power of Managing Uncertainty	Service Masters managed shared resources wit tentative work. This leads to uncertainty.
Schema, Contracts, and SLAs	It's all about the "black box" behavior of the service. How can we define this?
Wrapping Existing Apps with Biz Process	Humans make apps do long-running work. How can we make this driven by services more often
Layering for Business Process	We can build Service Agents just for managing biz process. This allows composable biz proces
Extrapolating Retail and Distributior	Looking at retail and distribution can tell us a lo about the composability of biz process.

Build Your Services Now!

- Microsoft Is Investing Lots in Services
  - Web Services Define Interop Standards
  - “Longhorn” Makes Service Development Easier
- You Can Build Services Now!
  - It’ll Be Easier with “Longhorn” but Don’t Wait!
  - Your Business Needs Services
  - Guidance Available to Help
    - http://msdn.microsoft.com/architecture
- Invest Now in Building Services!
  - Separate and Connect Your Apps As

Microsoft®